I T H E A

International Journal INFORMATION THEORIES & APPLICATIONS

2017 Volume 24 Number 2

International Journal

INFORMATION THEORIES & APPLICATIONS

Volume 24 / 2017, Number 2

International Journal "INFORMATION THEORIES & APPLICATIONS" (IJ ITA)
is official publisher of the scientific papers of the members of
the ITHEA International Scientific Society

IJ ITA welcomes scientific papers connected with any information theory or its application.
IJ ITA rules for preparing the manuscripts are compulsory.
The **rules for the papers** for IJ ITA are given on *www.ithea.org*.

Responsibility for papers *published in* IJ ITA belongs to authors.

International Journal "INFORMATION THEORIES & APPLICATIONS" Vol. 24, Number 2, 2017

Edited by the **Institute of Information Theories and Applications FOI ITHEA**, Bulgaria, in collaboration with:
Institute of Mathematics and Informatics, BAS, Bulgaria,
V.M.Glushkov Institute of Cybernetics of NAS, Ukraine,
Universidad Politécnica de Madrid, Spain,
Hasselt University, Belgium,
St. Petersburg Institute of Informatics, RAS, Russia,
Institute for Informatics and Automation Problems, NAS of the Republic of Armenia

Printed in Bulgaria
Publisher ITHEA®
Sofia, 1000, P.O.B. 775, Bulgaria. www.ithea.org, e-mail: office@ithea.org
Technical editor: **Ina Markova**

ISSN 1310-0513 (printed) **ISSN 1313-0463 (online)**

COMPARISON SOFTWARE SYSTEMS BASED ON INFORMATION QUALITY MEASURING

Krassimir Markov, Krassimira Ivanova, Stefan Karastanev

Abstract: *The usual analysis of experiments using rank-based multiple comparison was discussed in [Ivanova et al, 2016c]. In this paper we will outline another approach. It is based on the comparison of received results with user's information expectation, i.e. on quality of information about the systems received from experiments. All examples in the paper are based on results from real experiments presented in the [Markov et al, 2015].*

Keywords: *Quality of information, Evaluation of informational services; Rank-based multiple comparison.*

ITHEA Classification Keywords: *H.3.4 Systems and Software - Performance evaluation (efficiency and effectiveness); H.3.5 Online Information Services.*

Introduction

In the papers [Ivanova et al, 2016a; 2016b, 2016c] a method for comparison software systems was presented. In this paper we outline an extension of the method. It is a multiple comparison based of computing the quality of information received from the experiments. All examples in the paper are based on results from real experiments presented in the [Markov et al, 2015].

The formula for computing quality of information was published in [Markov et al, 1996a, 2006]. In this paper it will be used for ranging software systems. Firstly we will remember the main definitions concerning quality of information given in [Markov et al, 1996a, 2006]. After that we will outline the experiments and ranging based on Friedman test (ANOVA). Finally, ranging based on quality of information will be shown. A comparison of both approaches will be done in the conclusion.

Subjective information expectation and Quality of Information

Every entity which is active in respect to another entity, called "object" of this activity, is called "Subject" [Markov et all, 2006]. The Subject may reflect (temporary or permanently) a certain relationship from the object, i.e. the subject during its interaction with a particular entity (object) might reflect some of its elements and relations between them.

The reflection in the subject's consciousness which represents a real object is called "***Mental Information Model***" (MIM). The subject can establish a certain relationships between some of the mental information models in his conscious. In this case, the relationships form a set of interrelated MIM.

On the base of the already existing MIM, the subject forms (actively actual) mental model and turns to "**expect**" the connection of the new originated MIM with it.

The orientation towards (the origination of) inside-defined MIM, which depends on the concrete process of information interaction, is called subjective **"information expectation"** (IE). The types of IE were discussed in the paper [Markov et al, 1996b].

The Subject estimates the incoming information depending on the distance to the information expectation.

If the subject couldn't generate and include in his conscious such a "virtual" information model, we say there is no IE.

Quality of Information

The Subject combines the characteristics of the information expectation with ones of the incoming MIM. The combining the IE with some other MIM is called ***resolving the information expectation***.

Let "n" is the number of the characteristics of an information expectation. Some of them may be combined as well as the others could not. It is clear that "n" is always positive, i.e. n>0. If "n" is a zero then no IE exits.

When a new MIM is generated the Subject evaluates the distance between the IE and MIM. The more this distance is small, the more the IE is better resolved, i.e. satisfied and the incoming MIM is more qualitative.

Quality of the information (**Q**) is evaluated by the distance between the MIM and the IE (inverse proportional of distance between them).

It is proposed to compute the value of quality **Q** by the normalized formula [Markov et al, 1996a, 2006]:

$$\mathbf{Q = 1 / (1 + D),}$$

where **Q** is quality value; **D** is the distance between IE and MIM. The value of **D** depends on the types of information expectation [Markov et al, 1996b] and needs to be computed by corresponding formulas (see [Deza et al, 2012] or [Deza and Deza, 2016]).

For different types of IE we need different formulas for computing the distance R. For different goals of the subject the distance R may be defined as "linear" distance; as distance between corresponding "curves"; as distance between "subspaces"; etc. In this work we assume the simples case where the distance between IE and MIM is Euclidean.

Let remember that the **Euclidean distance** between points **p** and **q** is the length of the line segment connecting them. In **Cartesian coordinates**, if p = (p_1, p_2,..., p_n) and q = (q_1, q_2,..., q_n) are two points in Euclidean n-space, then the distance (d) from p to q, or from q to p is given by the Pythagorean formula:

$$D(p,q) = D(q,p) =$$

$$= \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + (q_3 - p_3)^2 + ... + (q_n - p_n)^2} =$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

Respectively, if sets of characteristics of IE and MIM are assumed as Cartesian coordinates, than we will have IE = (e_1, e_2,..., e_n) and MIM = (m_1, m_2,..., m_n) and Pythagorean formula:

$$D(IE, MIM) = D(MIM, IE) =$$

$$= \sqrt{(m_1 - e_1)^2 + (m_2 - e_2)^2 + (m_3 - e_3)^2 + ... + (m_n - e_n)^2} =$$

$$= \sqrt{\sum_{i=1}^{n} (m_i - e_i)^2}$$

Experiments

We had compared four real RDF-data storing systems: **R** [RDFArM, 2015], **V** [Virtuoso, 2013], **J** [Jena, 2016], and **S** [Sesame, 2015]. Systems **V**, **J**, and **S** are tested by Berlin SPARQL Bench Mark (BSBM) team and connected to it research groups [Becker, 2008; BSBMv2, 2008; BSBMv3, 2009]. System **R** was tested directly with the same data sets.

The experiments with middle-size RDF-datasets were based on selected real datasets from DBpedia [DBpedia, 2007a; 2007b] and artificial datasets created by BSBM Data Generator [BSBM DG, 2013; Bizer & Schultz, 2009]. The real middle-size RDF-datasets used consist of DBpedia's homepages and geocoordinates datasets with minor corrections [Becker, 2008]:

The artificial middle-size RDF-datasets, generated by BSBM Data Generator [BSBM DG, 2013], are published in N-triple as well as in Turtle format [BSBMv1, 2008; BSBMv2, 2008; BSBMv3, 2009]. We converted Turtle format in N-triple format using "rdf2rdf" program developed by Enrico Minack [Minack, 2010].

We have used four BSBM datasets – 50K, 250K, 1M, and 5M. Details about these datasets are summarized in following Table 1.

Table 1. Details about used artificial middle-size RDF-datasets

Name of RDF-dataset:	B50K	B250K	B1M	B5M
Exact Total Number of Instances:	50,116	250,030	1,000,313	5,000,453
File Size Turtle (unzipped)	14 MB	22 MB	86 MB	1,4 GB

Analysis of experiments: Rank-based multiple comparison

In [Ivanova et al, 2016c] we have presented experiments with middle-size and large RDF data sets, based on selected datasets from DBpedia's homepages and Berlin SPARQL Bench Mark (BSBM). The result from Rank-based multiple comparison is remembered below.

We had used the Friedman test to detect statistically significant differences between the systems [Friedman, 1940]. The Friedman test is a non-parametric test, based on the ranking of the systems on each dataset. It is equivalent of the repeated-measures ANOVA [Fisher, 1973]. We used Average Ranks ranking method, which is a simple ranking method, inspired by Friedman's statistic [Neave & Worthington, 1992]. For each dataset the systems are ordered according to the storing time measures and are assigned ranks accordingly. The best system receives rank 1, the second – 2, etc. If two or more systems have equal value, they receive equal rank which is mean of the virtual positions that had to receive such number of systems if they were ordered consecutively each by other.

Let **n** is the number of observed datasets; **k** is the number of systems.

Let r_j^i be the rank of system j on dataset i. The average rank for each system is calculated as

$$R_j = \frac{1}{n}\sum_{i=1}^{k} r_j^i .$$

The null-hypothesis states that if all the systems are equivalent than their ranks R_j should be equal. When null-hypothesis is rejected, we can proceed with the Nemenyi test [Nemenyi, 1963] which is used

when all systems are compared to each other. The performance of two systems is significantly different if the corresponding average ranks differ by at least the critical difference

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}$$

where critical values q_α are based on the Studentized range statistic divided by $\sqrt{2}$. Some of the values of q_α are given in Table 2 [Demsar, 2006].

Table 2. Critical values for the two-tailed Nemenyi test

quantity of systems	2	3	4	5	6	7	8	9	10
$q_{0.05}$	1.960	2.343	2.569	2.728	2.850	2.949	3.031	3.102	3.164
$q_{0.10}$	1.645	2.052	2.291	2.459	2.589	2.693	2.780	2.855	2.920

The results of the Nemenyi test are shown by means of critical difference diagrams.

Benchmark values from experiments are given in Table 3.

Table 3. Benchmark values

test / system	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
R	3	2272	14.79	3469	60	60	301	136412	1453	5901
S	3	2404	19	2341	179	213	1988	21896	44225	282455
V	2	1327	05	1235	23	25	609	7017	1035	3833
J	5	3557	13	3305	49	41	1053	70851	1013	5654

The ranking of the tested systems is given in Table 4.

Table 4. Ranks of tested systems

test / system	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	average rank
R	2.5	2	3	4	3	3	1	4	3	3	**2.85**
S	2.5	3	4	2	4	4	4	2	4	4	**3.35**
V	1	1	1	1	1	1	2	1	2	1	**1.2**
J	4	4	2	3	2	2	3	3	1	2	**2.6**

All average ranks are different. The null-hypothesis is rejected and we can proceed with the Nemenyi test. Following [Demsar, 2006], we may compute the critical difference by formula:

$$CD = q_{\alpha}\sqrt{\frac{k(k+1)}{6N}}$$

where $q\alpha$ we take as $q_{0.10}$ = 2.291 (from Table 1 [Demsar, 2006; Table 5a]); k is the number of systems compared, i.e. k=4; N is the number of datasets used in benchmarks, i.e. N=10. This way we have:

$$CD_{0.10} = 2.291 * \sqrt{\frac{4*5}{6*10}} = 2.291 * \sqrt{\frac{20}{60}} = 2.291 * 0.577 = 1.322$$

This way, we will use for critical difference $CD_{0.10}$ the value **1.322**.

At the end, average ranks of the systems and distance to average rank of the first one are shown in Table 5.

Table 5. Average ranks of systems and distance to average rank of the first one

place	system	average rank	Distance between average rank of the every system and average rank of the first one
1	**V**	1.2	**0**
2	**J**	2.6	**1.4**
3	**R**	2.85	**1.65**
4	**S**	3.35	**2.15**

The visualization of Nemenyi test results for tested systems is shown on Figure 1.

The order of the systems is (1) **V**, (2) **J**, (3) **R**, and (4) **S**.

Analyzing these experiments we may conclude that **R** is at critical distances to **J** and **S**.

R is nearer to **J** than to **S**.

R, **J**, and **S** are significantly different from **V**.

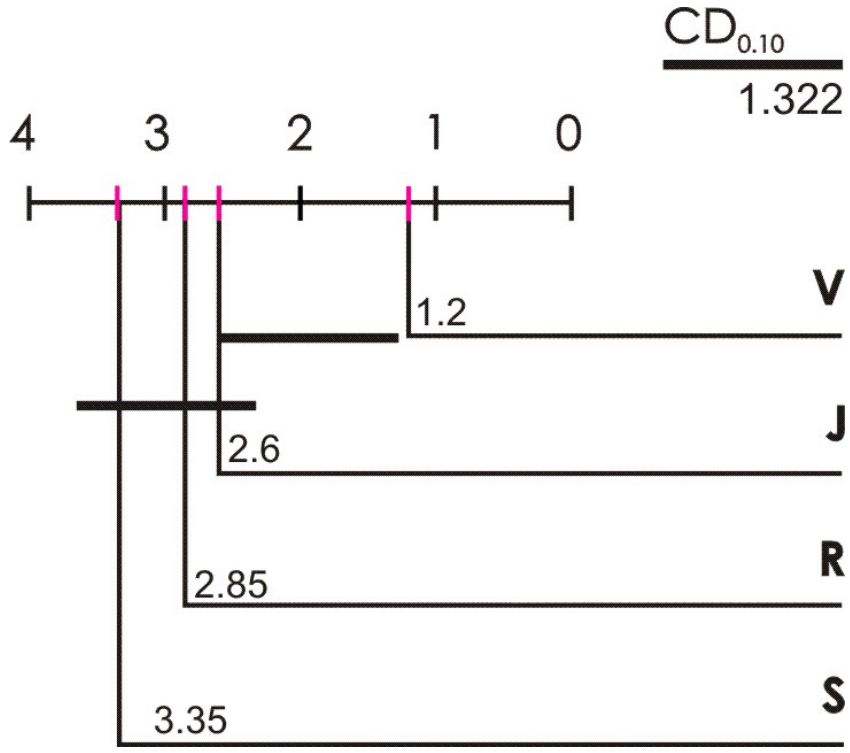


Figure 1. Visualization of Nemenyi test results

Comparison based on of the distance of experiments' information to the information expectation

What is important is that the Friedman test [Friedman, 1940] and ANOVA [Fisher, 1973] conceal the proportions and great differences between received data and this way the ranking does not take in account the distribution of data values. For instance, (see Table 3), in test 9 **S** is 42 times slower than **V**, and in test 10 **S** is 73 times slower than **V**, but in both cases it is on 4 place (see Table 4).

Below we will show another approach based on the distance to IE.

Firstly, we will transform data from Table 1 to be in the interval [0, 1] using transformation formula:

$$X_{new} = 1 - \frac{X_{old}}{MAX\_value\_of\_the\_test}$$

For instance, the results from Test 1 (second column of Table 6) will be transformed by formula:

$$X_{new} = 1 - \frac{X_{old}}{5}$$

because the worst storing time is 5 for the system **J**.

This transformation give us possibility to chose IE = (1, 1,..., 1)

Table 6. Transformed benchmark values and values of IE

test / MIM	1	2	3	4	5	6	7	8	9	10
IE	1	1	1	1	1	1	1	1	1	1
R	0.4	0.36125949	0.21052632	0	0.66480447	0.71830986	0.84859155	0	0.96714528	0.97910818
S	0.4	0.32414956	0	0.32516575	0	0	0	0.83948626	0	0
V	0.6	0.62693281	0.73684211	0.64398962	0.87150838	0.88262911	0.69366197	0.94856024	0.97659695	0.9864297
J	0	0	0.31578947	0.04727587	0.72625698	0.80751174	0.47032193	0.48061021	0.9770944	0.97998265

As we have pointed in previous sections, if sets of characteristics of IE and MIM are assumed as Cartesian coordinates, than we have IE = (e_1, e_2,..., e_n) and MIM = (m_1, m_2,..., m_n) and Pythagorean formula:

$$D(IE, MIM) =$$

$$= \sqrt{(m_1 - e_1)^2 + (m_2 - e_2)^2 + (m_3 - e_3)^2 + ... + (m_n - e_n)^2} =$$

$$= \sqrt{\sum_{i=1}^{n} (m_i - e_i)^2}$$

Using this formula, we compute distance between IE and MIM of every system (Table 7):

Table 7. Distance between IE and MIM of every system

MIM	Distance between IE and MIM
V	1.855536774
R	2.468421719
J	2.520168491
S	2.987382987

Finally, we compute the quality of information using formula:

$$Q = \frac{1}{1 + D(IE, MIM)}$$

Ranking of the systems based on quality of information for MIM of every system is given in Table 8.

Table 8. Ranking of the systems based on quality of information for MIM of every system

MIM	Q
V	0.350196856
R	0.288315574
J	0.284077311
S	0.250791059

Now we have new order of the systems (1) **V**, (2) **R**, (3) **J**, and (4) **S,** which takes in account data proportions.

The visualization of new results for tested systems is shown on Figure 2. The Critical Distance now is 0.049702899 or rounded off to 0.050 It is computed using formula:

$$CD = \frac{\max_Q - min_Q}{2}$$

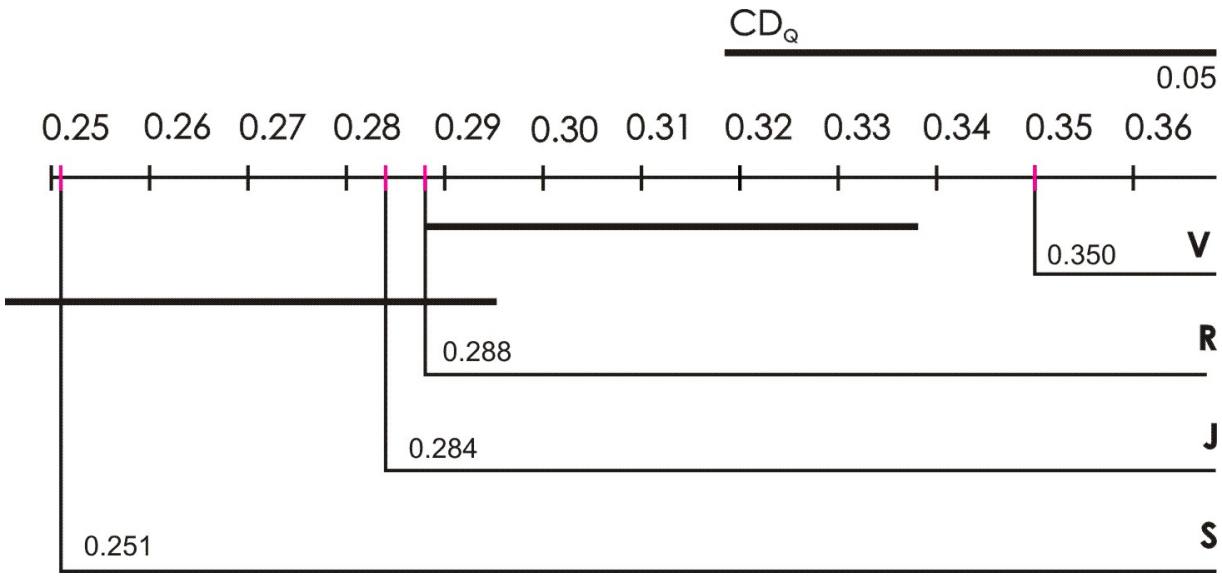


Figure 2. Visualization of quality test results

Analyzing these experiments we may conclude that R is at critical distances to J and S. R is much nearer to J than to S. R, J, and S are significantly different from V.

It is important that **R** and **J** change their places. Now **R** is at the second place.

Conclusion

We have presented results from series of experiments which were needed to estimate the storing time of four systems for middle-size and very large RDF-datasets. Experiments were provided with both real and artificial datasets. Experimental results were systematized in corresponded tables.

The main goal of this work was to propose a new ranking approach based on ***quality of received information***. We have remembered the main theoretical results from [Markov et al, 1996a, 2006] and using examples from real experiments we have shown the new approach is more reliable because it takes in account the distribution of the data values.

Acknowledgement

This paper is published with partial support by the ITHEA ISS (www.ithea.org).

Bibliography

[Becker, 2008] Christian Becker, "RDF Store Benchmarks with Dbpedia", Freie Universität Berlin, 2008, http://wifo5-03.informatik.uni-mannheim.de/benchmarks-200801/ (accessed: 05.04.2013)

[Bizer & Schultz, 2009] Christian Bizer, Andreas Schultz, "The Berlin SPARQL Benchmark", In: International Journal on Semantic Web & Information Systems, Vol. 5, Issue 2, Pages 1-24, 2009, http://wifo5-03.informatik.uni-mannheim.de/bizer/pub/Bizer-Schultz-Berlin-SPARQL-Benchmark-IJSWIS.pdf; see also http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/ (accessed: 31.07.2013)

[BSBM DG, 2013] Data Generator and Test Driver, In: Berlin SPARQL Benchmark (BSBM) - Benchmark Rules, http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/spec/BenchmarkRules/index.html#datagenerator (accessed: 31.07.2013)

[BSBMv1, 2008] Berlin SPARQL Benchmark Results, V1, 2008, http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/V1/results/index.html (accessed: 31.07.2013)

[BSBMv2, 2008] Berlin SPARQL Benchmark Results, V2 2008, http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/results/V2/index.html (accessed: 31.07.2013)

[BSBMv3, 2009] Berlin SPARQL Benchmark Results, V3, 2009, http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/results/V3/index.html (accessed: 31.07.2013)

[DBpedia, 2007a] DBpedia dataset "homepages.nt" dated 2007-08-30, http://wifo5-03.informatik.uni-mannheim.de/benchmarks-200801/H.nt.gz (accessed: 31.07.2013)

[DBpedia, 2007b] DBpedia dataset "geocoordinates.nt" dated 2007-08-30, http://wifo5-03.informatik.uni-mannheim.de/benchmarks-200801/G.nt.gz (accessed: 31.07.2013)

[Demsar, 2006] Demsar J, "Statistical comparisons of classifiers over multiple data sets", J. Mach. Learn. Res., 7, 2006, pp. 1 30.

[Deza and Deza, 2016] Michel Marie Deza, Elena Deza. Encyclopedia of Distances. Springer-Verlag Berlin Heidelberg, 2016. eBook ISBN: 978-3-662-52844-0; Hardcover ISBN: 978-3-662-52843-3; DOI: 10.1007/978-3-662-52844-0, Edition: 4, pp: 756

[Deza et al, 2012] Michel Deza, Michel Petitjean, Krassimir Markov. Mathematics of distances. ITHEA® 2012, Sofia, Bulgaria, ITHEA IBS ISC No.: 25. ISBN: 978-954-16-0063-4 (printed), ISBN: 978-954-16-0064-1 (online). 189 pp.

[Fisher, 1973] R. A. Fisher, "Statistical methods and scientific inference", (3rd edition) Hafner Press, New York, 1973, ISBN 978-002-844740-7.

[Friedman, 1940] Friedman, M., "A comparison of alternative tests of significance for the problem of m rankings", Annals of Mathematical Statistics, Vol. 11, 1940, pp.86-92

[Ivanova et al, 2016a] Krassimira Ivanova, Emiliya Saranova, Krassimir Markov, Stefan Karastanev, "A Method for Evaluation of Informational Services - Step 1: Computing the Hardware Proportionality Constants", International Journal "Information Technologies & Knowledge" Volume 10, Number 2, 2016, ISSN 1313-0455 (printed), ISSN 1313-048X (online). pp. 103- 110.

[Ivanova et al, 2016b] Krassimira Ivanova, Ivan Ivanov, Mariyana Dimitrova, Krassimir Markov, Stefan Karastanev, "A Method for Evaluation of Informational Services - Step 2: Computing the Informational Services' Performance Proportionality Constants", International Journal "Information Models and Analyses" Volume 5, Number 3, 2016, ISSN 1314-6416 (printed), ISSN 1314-6432 (Online)pp. 203 – 214.

[Ivanova et al, 2016c] Krassimira Ivanova, Krassimir Markov, Stefan Karastanev. A Method for Evaluation of Informational Services - Step 3: Rank-Based Multiple Comparison. International Journal "Information Content and Processing", Vol.3, Number 4, 2016, ISSN 2367-5128 (printed), ISSN 2367-5152 (online), pp. 303-321. (in print)

[Jena, 2016] Apache Jena, https://jena.apache.org/about_jena/about.html (accessed: 23.02.2016)

[Markov et al, 1996a] K.Markov, K.Ivanova, I.Mitov. Mental Information Measure. IJ ITA, 1996; v.4, n.1; pp. 11-16.

[Markov et al, 1996b] K.Markov, K.Ivanova, I.Mitov. Types of Information Expectation. IJ ITA, 1996; v.4, n.3, pp .21-24.

[Markov et al, 2006] Kr. Markov, Kr. Ivanova, I. Mitov. Basic Structure of the General Information Theory., pp. 19 32. IJ ITA, Vol.14, No.: 1, 2006. pp. 5-19.

[Markov et al, 2015] Krassimir Markov, Krassimira Ivanova, Koen Vanhoof, Vitalii Velychko, Juan Castellanos, „Natural Language Addressing", ITHEA® Hasselt, Kyiv, Madrid, Sofia, IBS ISC No.: 33, 2015, ISBN: 978-954-16-0070-2 (printed), ISBN: 978-954-16-0071-9 (online), 315 p

[Minack, 2010] Enrico Minack, "RDF2RDF converter", http://www.l3s.de/~minack/rdf2rdf/ 2010, (accessed: 31.07.2013).

[Neave & Worthington, 1992] Neave, H., Worthington, P., "Distribution Free Tests", Routledge, 1992

[Nemenyi, 1963] Peter Nemenyi, "Distribution-free multiple comparisons Unpublished", PhD thesis; Princeton University Princeton, NJ, 1963

[RDFArM, 2015] Krassimira Ivanova. RDFArM - A System For Storing Large Sets Of Rdf Triples And Quadruples By Means Of Natural Language Addressing. International Journal "Information Models and Analyses", Vol. 3, Number 4, 2014, ISSN 1314-6416 (printed), 1314-6432 (online), pp. 303 - 322.

[Sesame, 2015] Sesame, OpenRDF, https://bitbucket.org/openrdf/sesame (accessed: 01.12.2015)

[Virtuoso, 2013] OpenLink Virtuoso Universal Server: Documentation, http://Virtuoso.openlinksw.com/ (accessed: 23.11.2015)

Authors' Information

Krassimir Markov *– Institute of Mathematics and Informatics, BAS; ITHEA Institute of Information Theories and Applications, Bulgaria; e-mail: markov@foibg.com*

Major Fields of Scientific Research*: General theoretical information research, Multi-dimensional information systems; Business Informatics, Data Mining.*

Krassimira Ivanova *– Assoc. prof. Dr.; University of Telecommunications and Posts, Sofia, Bulgaria; Institute of Mathematics and Informatics, BAS, Bulgaria; e-mail: krasy78@mail.bg*

Major Fields of Scientific Research*: Software Engineering, Business Informatics, Data Mining, Multidimensional multi-layer data structures in self-structured systems*

Stefan Karastanev *– Assist. prof.; Institute of Mechanics, BAS, Bulgaria; e-mail: stefan@imbm.bas.bg*

Major Fields of Scientific Research*: Software Engineering, Data Processing and Mining, Data structures in information systems.*

APPLICATION OF BIOSENSORS FOR PLANTS MONITORING

Oleksandr Palagin, Volodymyr Grusha, Hanna Antonova,
Oleksandra Kovyrova, Vasyl Lavrentyev

Abstract: *Current methods of diagnostics of plant state need to conduct expensive and long-time physical-chemical and microbial analyses of soil and plant samples. The chlorophyll fluorescence induction method allows determining the functional state of plant in express mode without plant damage and it gives an opportunity to estimate the influence of stress factors on the plant state. In recent decades a number of researches of the chlorophyll fluorescence induction were significantly increased because of appearance of relatively inexpensive portable fluorometers. This paper represents results of testing biosensors developed at the V.M. Glushkov institute of Cybernetics of NAS of Ukraine on base of chlorophyll fluorescence induction method. It were developed appropriate software to facilitate data acquisition and processing. Analysis of some experimental results by means of neural networks is discussed.*

Keywords*: fluorometer, biosensor, wireless sensor network, chlorophyll fluorescence induction, neural network, information technology.*

ACM Classification Keywords: *H.4 Information system application*

Introduction

Over the past decade portable devices of "Floratest" family were developed and manufactured in V.M. Glushkov Institute of Cybernetics of NAS of Ukraine. The researchers of chlorophyll fluorescence induction (CFI) effect encounter the problem to gain sufficient amount of data by means of autonomous fluorometers. Besides, the time of a measurement of chlorophyll fluorescence induction varies from several minutes to one hour, depending on environmental conditions, species of plants and experiment specificity. The temperature and humidity of air and soil, illuminance can vary, that can influence on reliability of measuring data. All this has to be taken into account during ecological and agro-ecological monitoring. So, to overcome above-mentioned disadvantages, it was designed wireless biosensors that are combined in wireless sensor network together with special network coordinators, and concentrator [Palagin at al., 2017]. The biosensors were tested in laboratory and field conditions. This paper represents some important results of that testing and data analysis.

Work objectives

Work objectives are testing of developed biosensors and developing database, software and methods to facilitate acquisition and processing of measured data.

Measurement of CFI and its parameters

The technique of laboratory or field experiment includes next:

1. Selecting plants. Planning and choosing testing plants. Goal of an experiment has to be taken into consideration when experiment is planned and plants are selected. A chosen plant-indicator has to be sensitive to stress factor [Guo and Tan, 2015].

2. Plants are grown in identical conditions in pots or on field with identical soil.

3. The grown plants are divided into few groups – control and experimental.

4. Experimental plants are put on influence of stressful factors of different degree in accordance with testing program.

5. Network of biosensors measures chlorophyll fluorescence induction (CFI) of control and experimental plants in accordance with testing program for type of stress and its degree in scheduled terms.

The using of few fluorometers or the developed network of wireless biosensors allows reducing the time needed for measurements and it can provide data that are more adequate. The time can be calculated according to formula:

$$t_e = \frac{\sum_i^N (t_{ad} + t_m + t_{pr})}{N_s},$$

where t_e is a time to get experimental data; N is an amount of measurements; t_{ad} is a time of dark adaptation of leaf; t_{pr} is a time needed to prepare the next measurement; t_m is a time of measurement of chlorophyll fluorescence induction curve, N_s is a number of sensors.

If the sensors are placed on a leaf under sunlight then the dark adaptation has to be not less than 20 minutes. If the plant (or its leaf) is placed during long period in a shadow then 5 minutes is enough for the dark adaptation.

6. Measuring data of chlorophyll fluorescence induction, acquires by biosensors from control and experimental plants.

7. It is useful to record the air and soil temperature and humidity during a measurement of chlorophyll fluorescence induction. In addition, chemical and biological analysis of soil can be used for specific

biological researches. It is allows to take into consideration climatic effect as additional stress factor on parameters of chlorophyll fluorescence induction.

8. The results of measurements are processed by means of graphical, statistical and correlation analysis and machine learning technique. Before analysis, the measured data can be normalized.

9. The finish result of testing is detecting the sensibility of biosensors to influence of different stresses.

Typical curve of chlorophyll fluorescence induction is shown on figure 1. For analysis of measured curves the researchers typically analyze special parameters of CFI curves such as: F_o (initial level of chlorophyll fluorescence); F_m (maximum level of chlorophyll fluorescence); F_{st} (stationary level of chlorophyll fluorescence); $F_v = F_m - F_o$ (variable fluorescence); Fv/Fm ratio, Area (the area above the fluorescence curve between F_o and F_m), F_j (fluorescence value at point J, $t \approx 2$ ms); F_i (fluorescence value at I, $t \approx 30$ ms) and so on. Also the machine learning method is getting popular recent years [Kalaji at al, 2017].

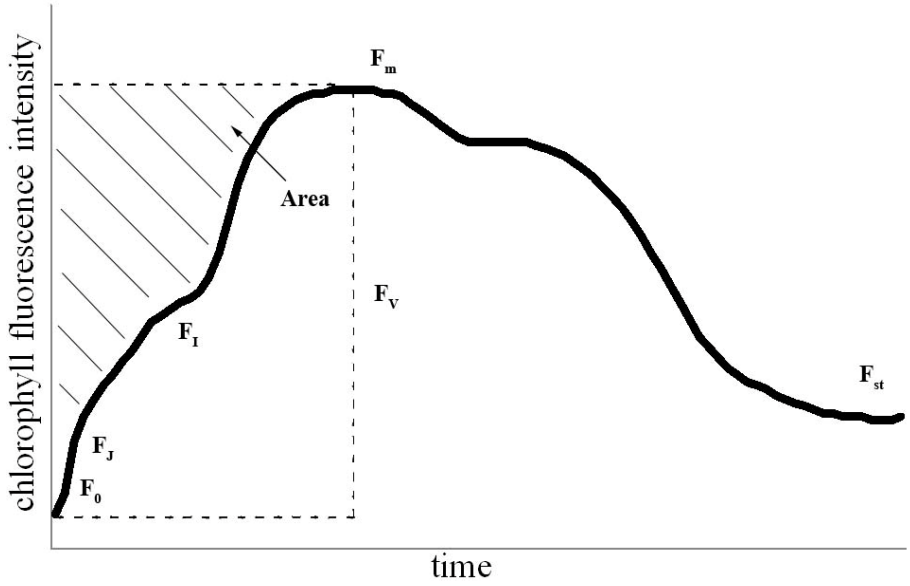


Figure 1. The typical curve of chlorophyll fluorescence induction

Development of software and database for work with chlorophyll fluorescence induction curves

Several activities have to be repeated during processing chlorophyll fluorescence induction curves (CFI) by means of personal computer: opening file with measuring results, graph building for previous visual estimation of dynamics of CFI curve, grouping different measurements, calculation of curves parameters and so on. It gets a lot of time. The special software FAnalyzer was developed to simplify the processing of chlorophyll fluorescence induction curves.

Functions of the developed software are the following:

1) Receiving measuring data from biosensors and further data output in form of a graph.

2) Storing the received measuring data on hard disk and opening in form of graphs;

3) Opening and storing several curves of CFI in one file. The file can be opened later and processed by means of program packages such as R, Excel, Matlab and so on.

4) Calculation and storing CFI curves parameters, that are frequently used to analyze the measuring results (F_m, F_o, F_{st}, Rfd, Area, F_i, F_j and other), and main statistical indicators for that parameters.

The graphical user interface of program is shown on Figure 2.

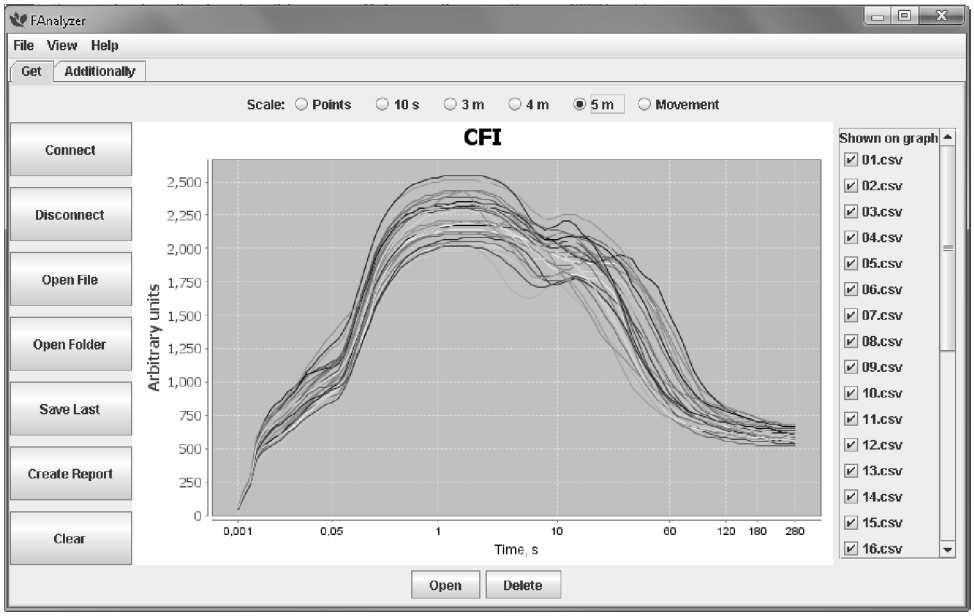


Figure 2. Graphical user interface

The suggested software allows to reduce time for preparing data to analysis, to calculate main parameters of CFI and proper statistical indicators. The calculation can be used for comparative analysis of plant states in conditions of influence of stress factors and in normal conditions.

During using multiple biosensors simultaneously it is necessary to store, process and visualize a large amount of measuring data. For convenience of users, the database and proper graphical user interface were developed. They allow storing a large amount of measurements in one place for further data analysis of measuring data by means of tools and methods, selected by user. During the database development, a set of entities was defined to represent in the database. The last ones contain information about: plant information; type of monitoring of plant state; measured curve of chlorophyll

fluorescence induction; information about soil, air and parts of plant (in case of chemical-biological analysis);information about devices and sensors, used for measurements; weather information; information about a person, conducting measurements; information about an organization and a location, where measurement was conducted.

A database management system MySQL was used for database implementation. The database diagram is shown on Figure 3.

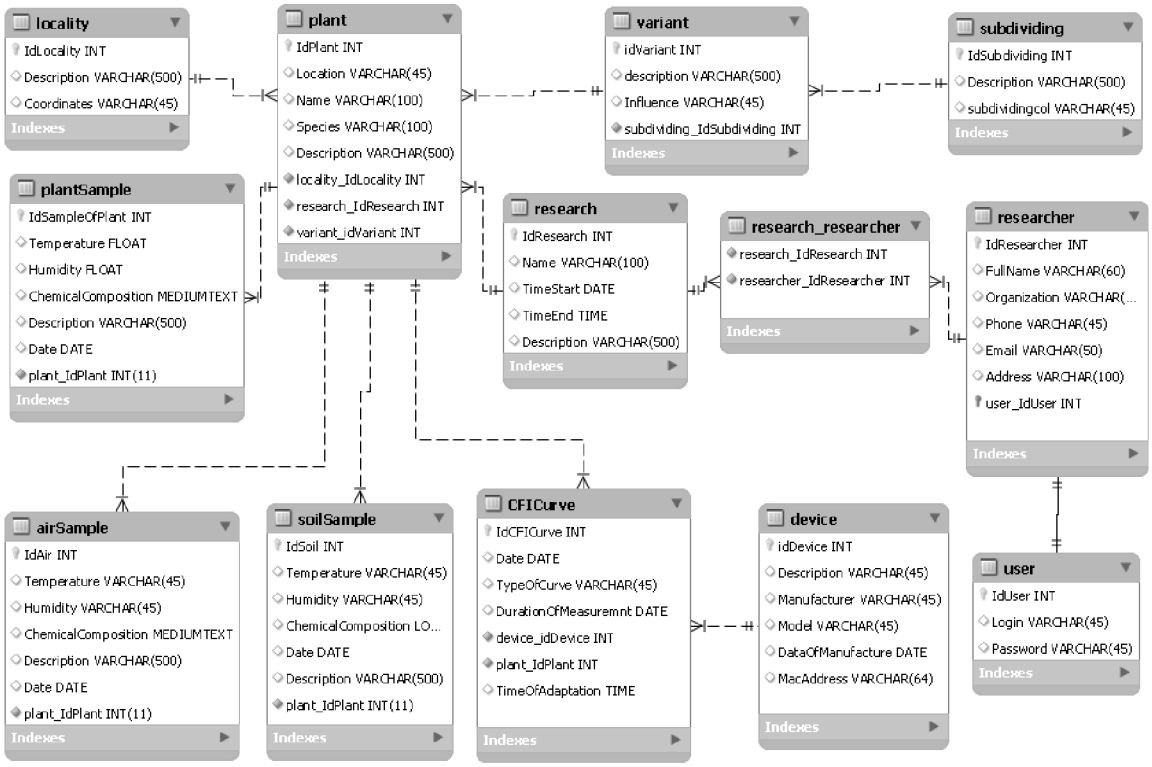


Figure 3. The database diagram

Importing data to the database can be carried by means of special software.

Research of change of chlorophyll fluorescence induction under influence of copper

To research the influence of heavy metals on plants it is reasonable to select the goose-foot plant. Goose-foot has a wide natural habitat, grows in a different environmental conditions. It was studied the influence of different doses of toxicant, copper sulphate ($CuSO_4$), on the test plants Plants were cultivated in 12 pots, three-four plants per pot. The plants were divided into 4 groups. Different concentration of $CuSO_4$ were dissolved in water and brought into the soil of these four groups.

Group 1 (V1) – control group without $CuSO_4$.

Group 2 (V2) – 1 g of CuSO4 / 1 kg of soil.

Group 3 (V3) – 3 g of CuSO4 / 1 kg of soil.

Group 4 (V4) – 6 g of $CuSO_4$ / 1 kg of soil.

The experiment was conducted during 13 days. At the beginning of the experiment the chlorophyll fluorescence induction was measured in all groups of plants (Figure 4). The same day the water solution of CuSO4 was brought into soil of test plants.

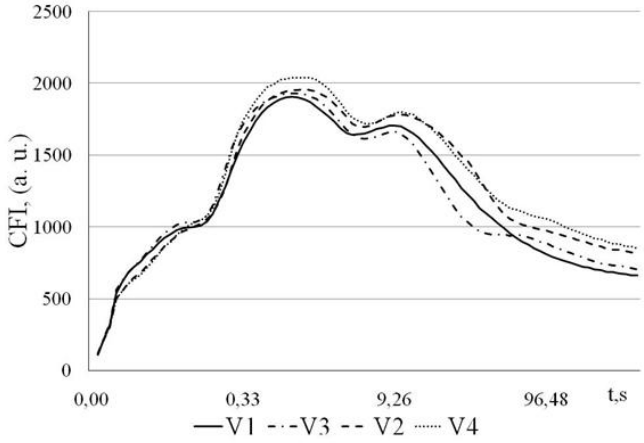


Figure 4. The intensity of chlorophyll fluorescence of goose-foot plant before of toxicant bringing in

The chlorophyll fluorescence intensity of test plants changed under the influence of toxicant. Figures 5 and 6 show graphs of the chlorophyll fluorescence on the second and third days of the impact of copper sulphate. It can be easily seen, that on the third day the maximum level of the chlorophyll fluorescence induction parameters (F_{st}, F_m) considerably decreased for plants that had been treated by toxicant.

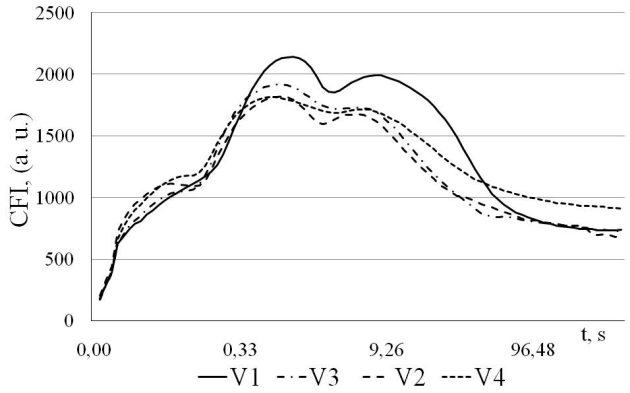


Figure 5. The intensity of chlorophyll fluorescence of goose-foot plant on the second day of toxicant influence

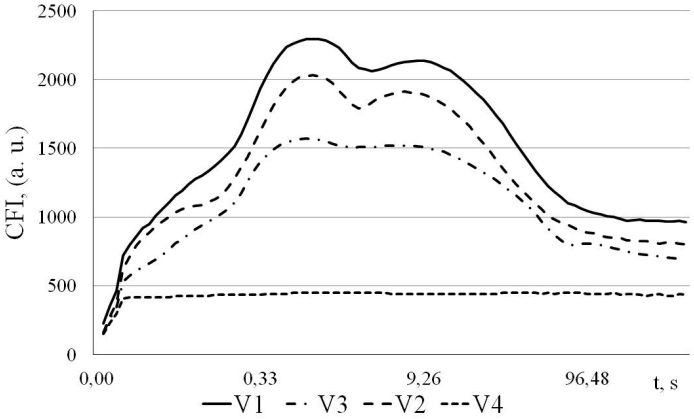


Figure 6. The intensity of chlorophyll fluorescence of goose-foot plant on the third day of toxicant influence

It should be noted, that on the sixth day of toxicant influence only one group of plants V2 remained, two other groups of plants V3 and V4 perished.

Analysis of parameter F_v/F_m provides information about the photochemical reactions, which are most sensitive to environmental factors. The maximum difference of the parameter F_v/F_m between the control group and the group V4, which received the maximum dose of copper sulphate, equals 38 %. At the beginning of the experiment this parameter had almost the same value in the three groups V1, V2, V3, V4 – 0,906 on the average. In the control group parameter F_v/F_m decreased by 5,8 % in comparison with the first day of measurement. In the group of plants V2 on the fifth day of toxicant influence the parameter F_v/F_m decreased by 5 % and the overall decrease equaled 4,6 % in comparison with the first day of experiment. The value of parameter in the group V3 decreased on the six day of the influence of copper sulphate. In the group V4 this parameter decreased by 39 % on the third day of the influence of toxicant. Figure 7 shows changes of the parameter during experiment.

Analysis of results shows, that different doses of copper sulphate influenced on the photosynthetic apparatus of plants in different ways. Thus, the dose of 6 grams of copper sulphate was critical for plant of group V4. Also, the dose of 3 g of copper sulphate is critical for plants of group V3 and causes irreversible changes in the plants. Photosynthetic apparatus of plants, treated by 3 g of $CuSO_4$, stops to function on eighth day of toxicant influence. The dose of 6 grams breaks the photosynthetic processes in plants on the third day. However, it should be noted, that the dose of 1 gram of $CuSO_4$ does not cause any serious changes in plants and also does not break the photosynthesis of plants.

For developing methodical support for wireless biosensors the experiment was conducted to research the influence of heavy metals on plants. It allowed to estimate the dose of copper sulphate, that is critical for plants, and to determine informative parameters of chlorophyll fluorescence induction curves.

During application of industrial methods the obtained results will be used to detect the presence of heavy metals in plants and estimate their impact on ecological state of certain territories.

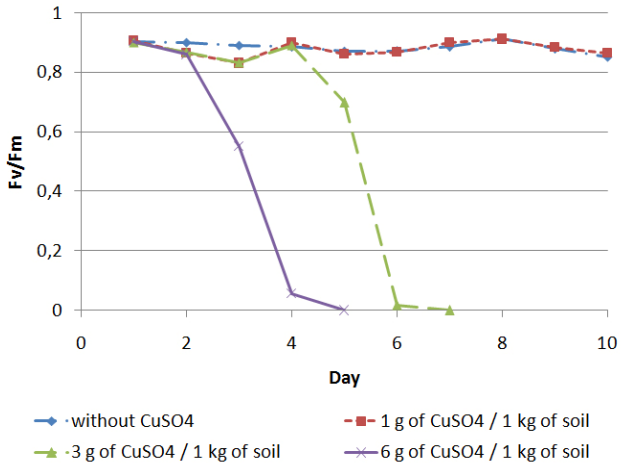


Figure 7. Changes of the parameter F_v/F_m during the testing

Using neural networks for determination of plants under stress

Nowadays neural networks and widely used for the analysis of biological and agricultural data in viral diseases of plants, pest determination, water consumption estimation, plant quality estimation etc. [Samborska at al.].

Researches of influence of herbicide on chlorophyll fluorescence were conducted at the V.M. Glushkov Institute of Cybernetics and the enough amounts of data were gained for using neural networks. Herbicide Roundup (glyphosate) was used for experiments. Roundup is a broad-spectrum systemic herbicide. The plants of Datura stramonium (weed) were divided into three groups. One, control group was not treated and two others were sprayed with different doses of herbicide.

Two-layer feed-forward network was chosen for classification of curves. Neural network has 89 inputs and 3 outputs (every measured curve consist of 89 points). Second, output layer consist of three neurons (three variants of curves). The required number of neurons of hidden layer was determined by conducting series of experiments. The performance (P) of the training was evaluated using means square error.

There were trained neural networks with different number of hidden layer neurons (from 1 to 364). The training of every network was repeated 30 times and the results were averaged and combined in vector P_{mean} (Figure 8). Thus, the neural network works most efficiently with not more than 70 neurons in the hidden layer. A neural network with 25 neurons in the hidden layer was chosen for further use. The

network uses the sigmoid transfer function for hidden neurons and the softmax function for output neurons.

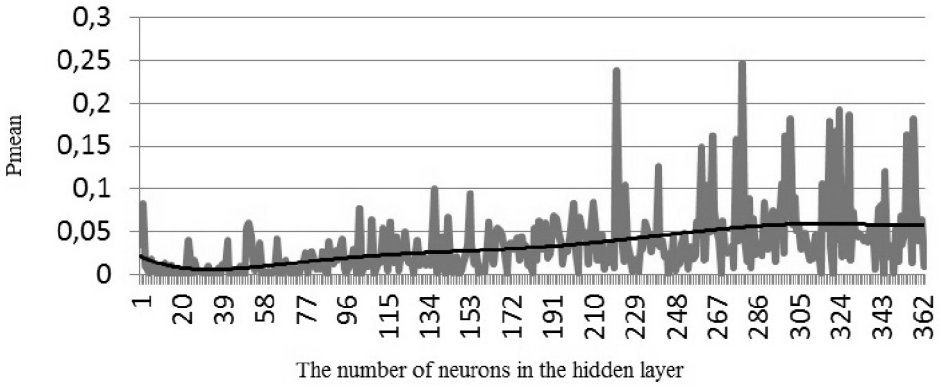


Figure 8. Dependence of the mean square error of the neural network training on the number of neurons

The neural network was trained on data measured in different days. The data measured in different days were used separately for training of the network. The results of the training are presented in Table 1.

As seen from Table 1, the smallest errors of recognition were received with data in 7 and 11 days. It is known that Roundup breaks the synthesis of the amino acids on 5-6 day and plants wade and discolor after two weeks. But after two weeks the curves of chlorophyll fluorescence of the treated leaves had serious difference even on one plant, therefore the neural network recognition is unsatisfactory. On the contrary the Student's test confirmed the difference between curves of plants of different groups at the end of second week.

Thus we showed that neural networks can be trained for stress recognition of plants using curves measured by sensors developed at V.M. Glushkov Institute of Cybernetics of NAS of Ukraine.

It is useful to use neural network for creating methods for evaluation of the state of plants in the city and the farm. It can be used at the stage of making decision (start watering, give fertilizer, etc.). Neural network training needs a representative set of data to make valid managerial decision, so the wireless biosensor networks allow to receive enough number of fluorescence induction curves.

Table 1. Results of the neural network training using data measured in different days, where E is an error of training, Ev is an error of validation, Et is an error of testing, Em is a mean calculated from three previous errors.

The number of curves	E, %	Ev,%	Et,%	Em,%	Notes
40	64,3	66,7	33,3	60,0	Before treatment of the herbicide
43	16,1	33,3	66,7	25,6	Before treatment of the herbicide
41	0	33,3	50,0	12,2	The third day after treatment
43	80,6	66,7	83,3	79,1	The fifth day
43	0	0	33.3	4.7	The seventh day
30	0	0	20	3.2	The eleventh
43	19,4	16,7	66,7	25,7	The thirteen
21	3,2	0	66,7	11,6	The twentieth

Using neural networks for determination of plant species

CFI curves of different plant species have some significant difference, thus they can be used for determination of specie of plant that are shown in [Kirova at al, 2009] by means of OJIP curve (CFI curve received during nearly 10 seconds) and neural network. With aim of testing the developed sensors for this task, a set of plants was measured. The set includes 176 curves from 6 species. The curves were measured during 5 minutes (full curve of CFI) and 10 seconds (OJIP curve) for next plants: soybean, goosefoot, ficus elastic, ficus benjamina, euphorbia, and zinnia.

Two-layer feed-forward network with 89 inputs and 25 neurons in the hidden layer was chosen. The network uses the sigmoid transfer function for hidden neurons and the softmax function for output neurons as in previous experiment. The output layer consists of 6 neurons.

The results of testing of the neural network present in Table 2. The neural network was trained 100 times and errors of testing were averaged after.

Table 2. The results of determination of plant species

Duration of measurement of CFI	5 minutes	10 seconds
minimal testing error, %	0	0
mean testing error,%	6,52	9,80

So, the curves of developed sensors can be used for taxonomic determination of plants. The curves measured during 5 minutes are more appropriate for this task. There are raised the issue of determination of plants with large amount of curves of very close species. The approach to solve it is described in [Kirova at al., 2009].

Conclusion

It was conducted the series of experiments for the testing biosensors developed at the V.M Glushkov Institute of Cybernetics of NAS of Ukraine to determine the sensitivity of biosensors to influence of stressful factors of different nature on experimental plants. The suitable software and database were developed to facilitate data processing. As result of using neural network, it can be concluded that neural network can recognize the different dose of fertilizer before changing of leaves appears and a 5 minutes measurement of CFI is more informative for determination of plant species then 10 seconds measurement.

Bibliography

[Guo and Tan, 2015] Y. Guo, J. Tan. Recent advances in the application of chlorophyll fluorescence from photosystem II. Photochemistry and Photobiology Vol. 91, Issue 1, Wiley, 2015. pp. 1-15

[Kalaji at al., 2017] H.M. Kalaji, G. Schansker, M. Brestic at al. Frequently asked question about chlorophyll fluorescence, the sequel. Photosynthesis Research. Vol. 132, Issue 1, Springer, 2017. pp 13-66.

[Kirova at al., 2009] M. Kirova, G, Ceppi, P. Chernev, V. Goltsev, R. Strasser Using artificial neural networks for plant taxonomic determination based on chlorophyll fluorescence induction curves.

Biotechnology & Biotechnological Equipment. Vol. 23, Issue sup1: XI Anniversary scientific conference, 2009 pp. 941-945. ISSN: 1310-2818 http://dx.doi.org/10.1080/13102818.2009.10818577

[Palagin at al., 2017] O. Palagin, V. Romanov, I. Galelyuka, O. Voronenko, Y. Brayko, R. Imamutdinova. Wireless sensor network for precision farming and environmental protection. I.Tech 2017

[Samborska at al., 2014] I. Samborska, V. Alexandrov, L Sieszko, B. Kornatowska, V. Goltsev, M.D. Cetner, H.M. Kalaji. Artificial neural networks and their application in biological and agricultural research. Signpost Open Access Journal NanoPhotoBioSciences, Vol. 02, 2014. pp. 14-30. ISSN: 2347-7342 http://signpostejournals.com/ejournals/portals/12/v22.pdf

Authors' Information

Oleksandr Palagin *– Depute-director of V.M.Glushkov Institute of Cybernetics of National Academy of Sciences of Ukraine, Academician of National Academy of Sciences of Ukraine, Doctor of technical sciences, professor; Prospect Akademika Glushkova 40, Kiev, 03187, Ukraine; e-mail: palagin_a@ukr.net*

Volodymyr Grusha *– research fellow of V.M.Glushkov Institute of Cybernetics of National Academy of Sciences of Ukraine; Prospect Akademika Glushkova 40, Kiev, 03187, Ukraine; e-mail: vhrusha@gmai.com; website: http://www.dasd.com.ua*

Oleksandra Kovyrova*– research fellow of V.M. Glushkov Institute of Cybernetics of National Academy of Sciences of Ukraine; Prospect Akademika Glushkova 40, Kiev, 03187, Ukraine; e-mail: kovyrova.oleksandra@gmail.com; website: http://www.dasd.com.ua*

Antonova Hanna *– engineer of V.M. Glushkov Institute of Cybernetics of National Academy of Sciences of Ukraine; Prospect Akademika Glushkova 40, Kiev, 03187, Ukraine; e-mail: annat7806@gmail.com; website: http://www.dasd.com.ua*

Vasyl Lavrentyev *– research fellow of V.M. Glushkov Institute of Cybernetics of National Academy of Sciences of Ukraine; Prospect Akademika Glushkova 40, Kiev, 03187, Ukraine; e-mail: vaslavr@i.ua; website: http://www.dasd.com.ua*

TOWARD MEASURING LINGUISTIC COMPLEXITY: GRAMMATICAL HOMONYMY IN THE RUSSIAN LANGUAGE

Olga Nevzorova, Alfiya Galieva, Vladimir Nevzorov

Abstract*: Currently linguistic complexity is one of the most debatable concepts in linguistics, and there are different ways of understanding this complexity depending on linguistic domains, research aims and theoretical background. We proceed from the assumption that linguistic complexity becomes apparent in those parameters that can be measured. Grammatical homonymy is an important manifestation of structural complexity of a language, and many aspects of it are computable.*

The study of grammatical homonymy from the point of view of linguistic complexity requires development of appropriate methodology. We examined this phenomenon on linguistic data of the extended version of A. Zaliznyak dictionary using the software of Ontointegrator system. We distinguished four structural parameters that enable to disclose statistical aspects of grammatical homonymy relevant for language processing. The distribution of grammatical homonyms manifests basic typological features of a language.

Keywords*: grammatical homonymy, linguistic complexity, the Russian language, word forms, parts of speech.*

ITHEA Keywords*: H.3.1 Content Analysis and Indexing.*

Introduction

The phenomenon of linguistic complexity (language complexity) became one of the topics of great importance in linguistics in the last decades. Various researches represent dissimilar ways of theoretical understanding of the phenomenon of complexity and propose different parameters of measuring and different ways of practical evaluation of this complexity ([Kusters, 2003; Dahl, 2004; Bane, 2008; Gil, 2008; Juola, 2008; Miestamo, 2008; Newmeyer, 2014; Becerra-Bonache, 2015] and other works).

Modern science worked out different approaches to determine the complexity of an object, and these approaches may be reduced to two basic types:

1) Complexity as a characteristic of objective (structural, dynamic and other) properties of the system;
2) Complexity as a characteristic of the process of cognition and studying an object, rather than of the system.

In [Rastrigin, 1981] the following basic properties of complex objects are specified:

1. Lack of necessary mathematical description.
2. "Noisiness" of complex systems which is evoked not by special generators of random hindrances, but rather by individual complexity of an object and by resulting inevitable abundance of secondary processes, so the object behavior seems in many cases unexpected to the researcher.
3. Intolerance to external control.
4. Non-stationarity of the complex system that shows up in drifting characteristics of the system, in changing its parameters, and in evaluation of the system in time.
5. Impossibility in many cases to reproduce the experiments, due to "noisiness" and non-stationarity of the complex system [Rastrigin, 1981].

The properties of complex objects named above, are generally applicable for characterizing natural languages, nevertheless, with certain provisos. In particular, experiments on natural language are irreproducible in the sense that results of analyzing diverse texts and diverse text collections may significantly differ, which is caused by complicated interaction of systemic, functional, individually authored and other factors. With respect to external influences, different subsystems of the language behave differently, so we can distinguish two types of these subsystems:

– open ones – vocabulary (languages easily accept new words), lexical semantics (words of a language get new senses);
– closed ones – from a synchronic viewpoint grammar is a closed system, because new grammatical categories and grammatical meanings hardly emerge.

Word formation may be regarded as a borderline domain: new words appear with ease, but as a rule, only in derivation models that are admissible for the language system itself.

Putting the question of linguistic complexity requires development of definitions and objective criteria of this complexity. Apparently, the degree of complexity may significantly differ depending on who would

assess the language, within what linguistic theory, on what layer data and what form (written or oral) of the language.

Studying the data of grammatical dictionaries and grammatically annotated corpora may be regarded as a tool for measuring quantitatively expressed parameters of linguistic complexity on the level of grammar. This paper is a first step to understanding the phenomenon of linguistic complexity basing on data on distribution of part of speech homonymy in Russian; the data is retrieved from the extended version of Grammatical Dictionary of A. Zaliznyak [Zaliznyak, 1987].

We aim to uncover certain formalized and measurable parameters of linguistic complexity; the main focus is on grammatical homonymy in the Russian language. The rest of the paper is organized as follows: Section 2 presents a brief description of related works. Section 3 defines main factors influencing linguistic complexity. Section 4 gives an analysis of part of speech distribution of homonymous word forms from the viewpoint of linguistic complexity. Section 5 concludes by summarizing main results and indicating future research.

Related works

Although the concept of linguistic complexity seems intuitively clear, it has scarcely undergone formalization and analysis. Researchers regard different criteria and parameters of linguistic complexity and get different, even opposite results for the same language.

A. Berdichevsky [Berdichevsky, 2012] gives an overview of approaches to theoretical understanding of language complexity and concludes that there are three main illations confirmed by most researches. First, commonly accepted ideas about equal complexity of all languages is not true. Not only can researchers rank languages by complexity, but they also aim at measuring the complexity of a language, or, at least, of a fragment of a language, using quantitative methods. At last, such measuring, as well as certain qualitative studies, illustrate that linguistic complexity is influenced by social factors [Berdichevsky, 2012].

The dissertation of W. Kusters *Linguistic Complexity. The Influence of Social Change on Verbal Inflection* [Kusters, 2003] investigates the influence of extralinguistic factors on internal language structure. The author studies verbal inflection in certain languages (Arabic, Scandinavian, Quechua and Swahili) and argues that a large number of non-native speakers of a language, social cohesion within a speech community, and enlargement of external contacts can lead to decreasing the complexity of verbal inflection.

In [Dahl, 2004] is represented methodologically significant delimitation of a number of essential concepts: *complexity, cost, difficult*y and *demandingness.* According to this researcher, *complexity* is a theoretical construct aimed at determining "objective' parameter of a language, important for language

processing that must not be related to a user or an agent. The notions of *cost* and *difficulty* are relevant for adult language learners. Cost implies essentially "the amount of resources – in terms of energy, money or anything else – that an agent spends in order to achieve some goal" [Dahl, 2004]. High cost does not necessarily imply high degree of complexity – the relationship between these phenomena is not direct. "Difficulty is a notion that primarily applies to tasks, and is always relative to an agent: it is easy or difficult for someone" [Dahl, 2004]. Demandingness is a link between complexity and difficulty: for instance, acquiring a human language natively is certainly demanding (only human children seem to fulfil the requirements), but it does not necessarily follow that children find it difficult [Dahl, 2004].

The paper of P. Juola [Juola, 2008] discusses some definitions proposed in literature, and shows how complexity can be assessed in various frameworks. The author focuses on mathematical and psychological aspects of complexity, and attempts to validate available complexity measurements.

We may say that the topic of complexity of languages has different dimensions and nowadays attracts a great deal of interest. Researchers maintain that language complexity may be regarded and evaluated on different levels: of the language as a whole, and of its separate layers; thus parameterisation of linguistic complexity needs further research, and work results must be considered in the general theory of language.

Parameters of complexity: toward a definition

The notion of complexity is conceptualised and defined differently in different domains. To specify this notion we are to take into consideration peculiarities of the internal organisation of the system, its evolution, interaction with the external world, etc. We are to realise that the actual diversity of internal relations of a complex object is not easy to merely describe and parameterise, but also to discover in many cases. That is essential for such a multidimensional phenomenon as language.

Assessment of linguistic complexity supposes search for objectively evaluating and finding comparable criteria. To determine the absolute value of complexity many researchers ([Dahl, 2004], [Juola, 2008] and other) use a categorical apparatus of information theory, and Kolmogorov complexity may serve as an example of that. Kolmogorov complexity may be defined as a way of measuring the amount of information in a given string – as the length of the shortest possible algorithm required to describe/generate that string [Juola, 2008]. Because of practical uncomputability and nonapplicability of Kolmogorov complexity for linguistic phenomena, P. Juola applies a purely technical expedient and considers file compression method as an attempt to approximate this kind of complexity within a tractable formal framework [Juola, 2008].

J. McWhorter, assessing linguistic complexity, relies upon the assumption that an area of grammar is more complex than the same area in another grammar to the extent that it encompasses more overt

distinctions and/or rules than another grammar [McWhorter, 2001]. This assumption is deployed in the following way:

1. A phonemic inventory is more complex to the extent that it has more marked members.
2. A syntax is more complex than another to the extent that it requires processing more rules, such as asymmetries between matrix and subordinate clauses.
3. A grammar is more complex than another to the extent that it gives overt and grammaticalized expressions to more fine-grained semantic and/or pragmatic distinctions than another.
4. Inflectional morphology renders a grammar more complex than another one in most cases [McWhorter 2001].

Reduction to a common denominator of great variety of grammatical phenomena of different languages remains an insoluble problem; nevertheless the first steps in this direction may be made by means of automatic text processing.

Grammatical homonymy from the linguistic complexity view point

Linguistic literature does not present similar views on homonymy. Disputable items are the content of the concept, the principles of classification and classification schemes. The most general classification distinguishes lexical homonyms which represent the same category of parts of speech, and grammatical homonyms which are related to different parts of speech. Grammatical homonymy is an important manifestation of structural complexity, and formal and quantitative characterization of homonymous structures within and across languages can provide a complexity ranking for them in many respects.

In this paper we consider grammatical (part of speech) homonymy in Russian on the data of grammatical dictionary of A. Zaliznyak. The work is aimed at examining statistical characteristics of grammatical homonymy and at eliciting complexity parameters of this phenomenon.

Investigation of statistical properties is carried out by means of *Ontointegrator* software system developed by O. Nevzorova and V. Nevzorov [Nevzorova, 2009]. As the linguistic data source we used the extended version of A. Zaliznyak dictionary that was deployed in the system as a paradigmatic list of words. Total volume of the dictionary is 133,040 lexemes (3,162,600 word forms). Each word form is coded by two numerical characteristics that define constant and variable grammatical characteristics of the word form, the latter depending on the part of speech. Each homonym is marked by two or more sets of grammatical characteristics.

Figure 1 shows the basic screen form of *Ontointegrator* system for work with a grammatical dictionary. The *Ontointegrator* system has the Russian interface. Figure 2 presents distribution of word forms by

parts of speech in a grammatical dictionary (by the time the article was written). Figure 3 displays distribution of words by parts of speech.

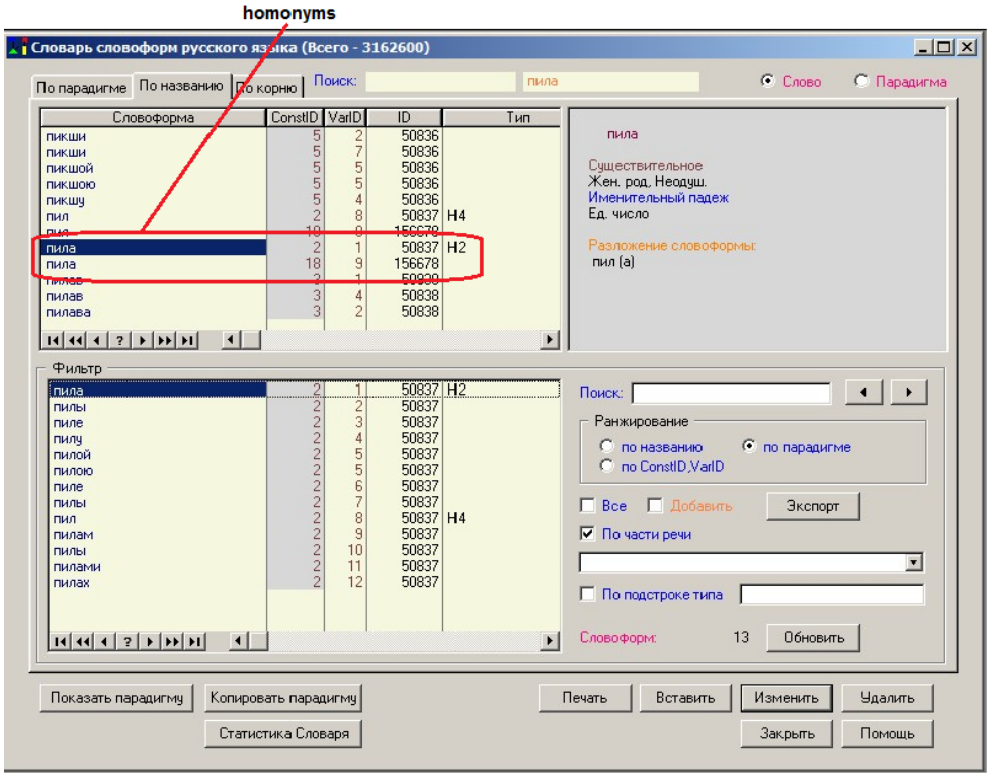


Figure 1. Basic screen form of *Ontointegrator* system for work with a grammatical dictionary

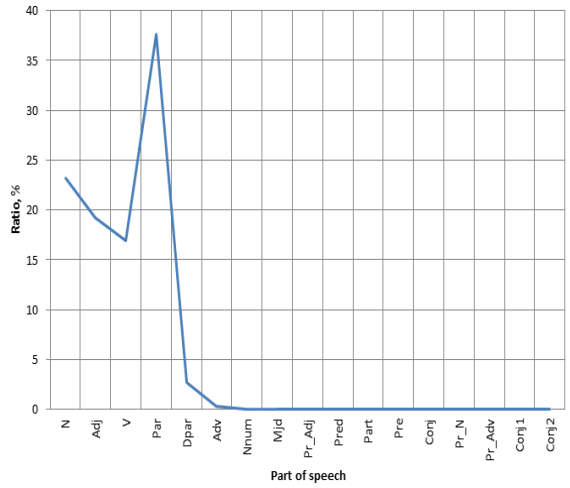


Figure 2. Distribution of word forms by parts of speech in a dictionary

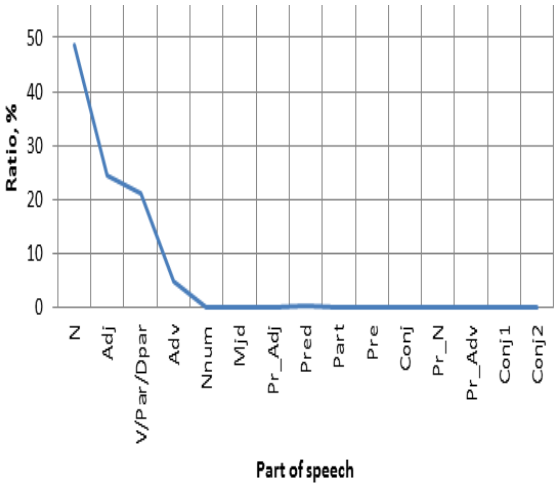


Figure 3. Distribution of words by parts of speech in a dictionary

For designating parts of speech the following abbreviations are used:

N – noun; ADJ – adjective; V – verb; PAR – participle; DPAR – the gerund; ADV – adverb; CONJ – conjunction; PRE – preposition; PART – particle; MJD – interjection; PRED – predicate word; NNUM – numeral; PR_ADJ – pronominal adjective; PR_N – pronominal noun; PR_ADV – pronominal adverb; CONJ1 – syndetic word of type 1; CONJ2 – syndetic word of type 2.

Based on grammatical characteristics of word forms we built statistical distributions by different characteristics to get a full picture of performance of grammatical homonymy in Russian.

Figure 4 displays distribution of grammatical homonyms/non-homonyms within each part of speech. Figure 4 illustrates the contribution of homonyms into each part of speech (class), i.e. into total number of elements of a given part of speech. For instance, all elements of class Conj1 (in Russian: *chego, kogo, chem, chto, kom, komy, kem, chemy*) – 8 items in all) are grammatical homonyms; class ADV contains 75,1% of grammatical homonyms, and in classes N, ADJ, V and PAR grammatical homonymy is imperceptible in relation to total number of members of these classes.

Proceeding from the analysis of distribution of homonyms within classes (parts of speech) we may identify *homonymity parameter* which may help us distinguish between strong (containing large percentage of homonyms) classes and weak (containing small percentage of homonyms) classes. 10%. is taken as a conditional threshold of division.

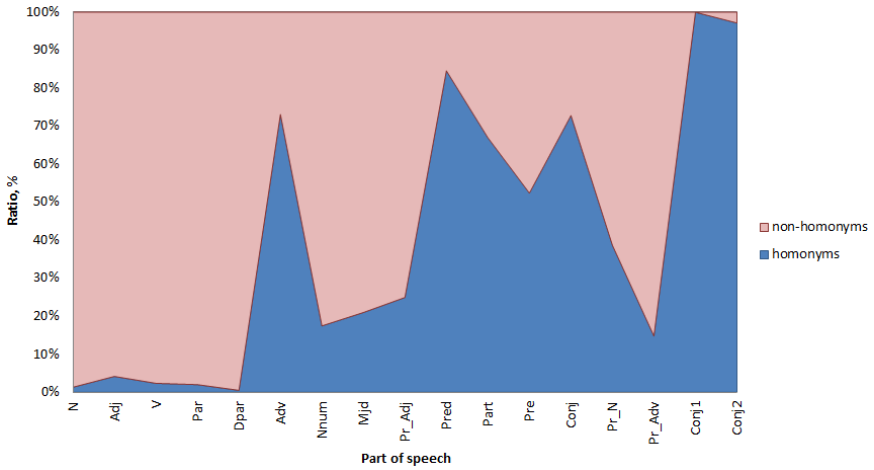


Figure 4. Distribution of grammatical homonyms/non-homonyms within each part of speech

Strong classes are ADV, NNUM, MJD, PR_ADJ, PRED, PART, PRE, CONJ, PR_N, PR_ADV, CONJ1, Conj2. Weak classes are N, ADJ, V, PAR, DPAR. Homonymity parameter reflects basic typological features of Russian morphology and syntax, where for example, homonymy of adverbs and predicate words, and of pronouns and syndetic words is a stumbling-block for disambiguation.

Figure 5 shows distribution of grammatical homonyms/non-homonyms of each part of speech (contribution to total volume of the dictionary); for visual assessment of correlation between them the amount of homonyms is displayed with the scale factor of 10. Figure 6 displays the same distribution for strong classes on an enlarged scale.

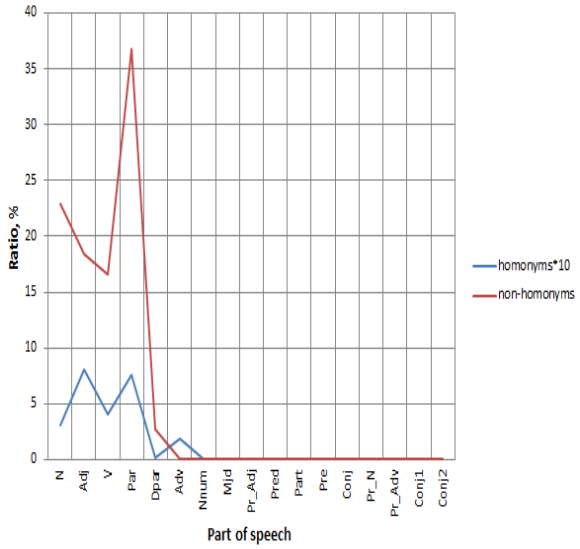


Figure 5. Distribution of grammatical homonyms/non-homonyms of each part of speech

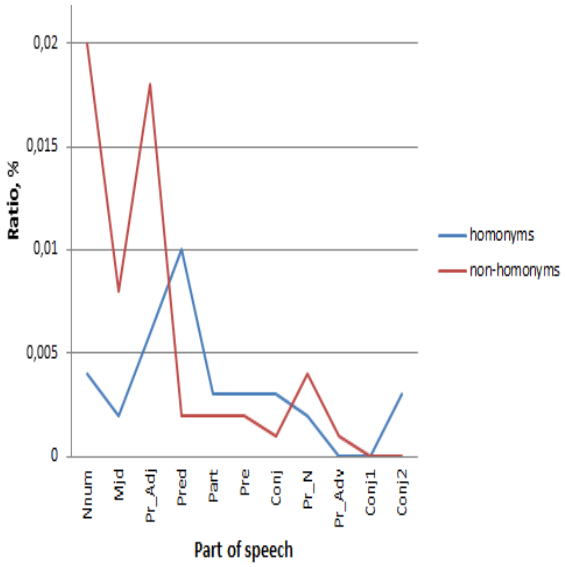


Figure 6. Distribution of grammatical homonyms/non-homonyms of each part of speech (on an enlarged scale) for strong classes

Another characteristics of the homonymy system of a language is the *power of types of grammatical homonymy*. The value of this parameter determines the number of different combinations of part of speech categories (classes) in the set of all word forms of the language. The Russian language by power of types of grammatical homonymy has value 126. The parameter of power of types of grammatical homonymy gives us the absolute numeric value of types of homonymy fixed in the dictionary.

Table 1 represents data on all types of grammatical homonymy and power of classes (estimated on belonging to the selected range of values).

The next step of the study was to examine the system of grammatical homonymy of the binary type, i.e. the object of study was grammatical homonyms with two characteristics. The first characteristic was fixed (given), and the second could vary depending on the class. In this way we built systems like Y/X, where Y and X were change within the spectrum of classes i.e. for Y=N this is N/ADJ, N/V, N/PAR, N/DPAR etc.

Table 1. Distribution of types of grammatical homonymy

Types of grammatical homonymy	Number of types	Range of values of power of the type
ADJ/PAR, PAR/V	2	5000-10000
ADJ/ADV, ADJ/N, N/V	3	1000-4999
N/PAR	1	500-999
ADJ/N/PAR, DPAR/N, ADJ/V	3	200-499
ADJ/ADV/PRED, ADV/N	2	100-199
ADJ/ADV/V, ADJ/PAR/V, ADJ/ADV/PAR, CONJ2/PR_ADJ, N/NNUM	5	50-99
ADJ/DPAR, ADJ/N/V, ADJ/PRED, ADV/PRE, CONJ/PART, MJD/N, PR_ADJ/PR_N	7	20-49
ADJ/PR_ADJ, ADV/CONJ, ADV/NNUM, ADV/PRED, ADV/V, N/PRED, N/PR_ADJ, N/PAR/V,	8	10-19
ADJ/ADV/N, ADJ/ADV/PRE, ADV/DPAR, ADV/CONJ/PART, ADV/PART, CONJ2/V, DPAR/PRE, N/PART, N/PRE, PART/V, PR_ADJ/V	11	5-9
ADJ/ADV/CONJ/PART, ADJ/ADV/N/PRED, ADJ/ADV/PART, ADJ/ADV/PART/PRED, ADJ/DPAR/N, ADJ/N/PART, ADJ/N/PR_ADJ, ADJ/N/PAR/V, ADJ/PR_N, ADV/N/PART, ADV/N/PRE, ADV/N/V, ADV/PAR, ADV/PRED/V, CONJ/PART/PR_ADV, CONJ/V, CONJ/CONJ1/PR_N, CONJ1/N/PR_ADJ, CONJ1/PR_N, CONJ1/N/PR_N, CONJ2/N/PR_ADJ, DPAR/N/V, DPAR/PAR, DPAR/V, DPAR/PR_ADJ, MJD/PART, N/NNUM/PAR, NNUM/PAR/V, NNUM/PR_N, NNUM/PR_ADJ/PR_N, NNUM/V, PAR/PR_ADJ/V, PART/PR_ADJ/PR_N, PRED/V	34	2-4
Другие типы	50	1
Total number of types	**126**	

Figure 7 shows distribution of the system of binary homonyms N/X with respect to all homonyms from N. Figure 8 represents this distribution on an enlarged scale.

To characterize the homonymy systems of binary type we entered third parameter - *binary expression* with ranking values (strong, average, weak). The binary expression parameter for the system N/X has average value (9 pairs from 17 are valuable). For comparison we provide Figure 9 and 10 displaying the system of binary homonyms of the PAR/X type. The binary expression parameter for this system is weak (6 pairs from 17 are valuable).

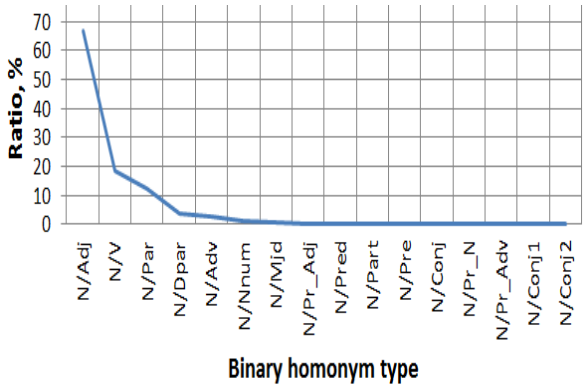


Figure 7. Distribution of binary homonyms of N/X type with respect to all homonyms from N

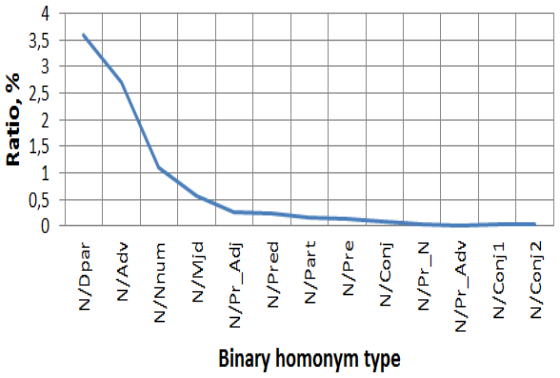


Figure 8. Distribution of binary homonyms of N/X type with respect to all homonyms from N (on an enlarged scale)

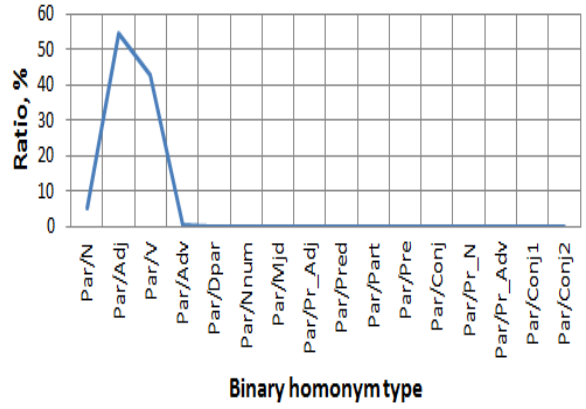


Figure 9. Distribution of binary homonyms of PAR/X type with respect to all homonyms from PAR

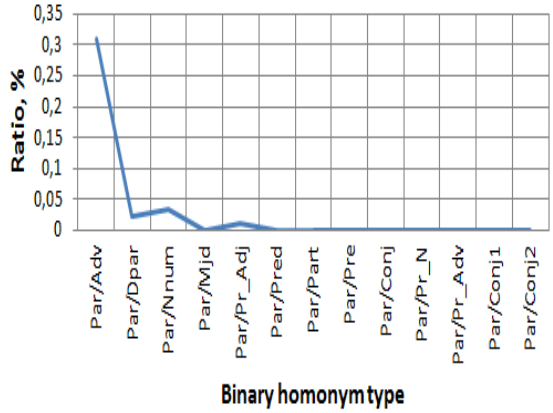


Figure 10. Distribution of binary homonyms of PAR/X type with respect to all homonyms from PAR (on an enlarged scale)

The fourth characteristics – *complication of binary parameter* for the system of binary homonyms is connected to the analysis of complication types in the structure of binary characteristics. For each binary parameter we obtained data on types of its complication, i.e. what additional classes (parts of speech) may extend the state of characteristics of the homonym. So we detected homonyms with 3 and 4 members, for example N/ADJ/V (in Russian: *zeleney, krylo*) or N/ADJ/ADV/PAR (in Russian: *gorychim*). Figure 11 presents the picture of complication for binary homonym N/ADJ. For this binary homonym complication parameter has high value (9 from 17) and in the structure of complication we find groups of homonyms of 3 members (6 groups) and 4 members (2 groups). The same plots are built for all binary parameters of all parts of speech.

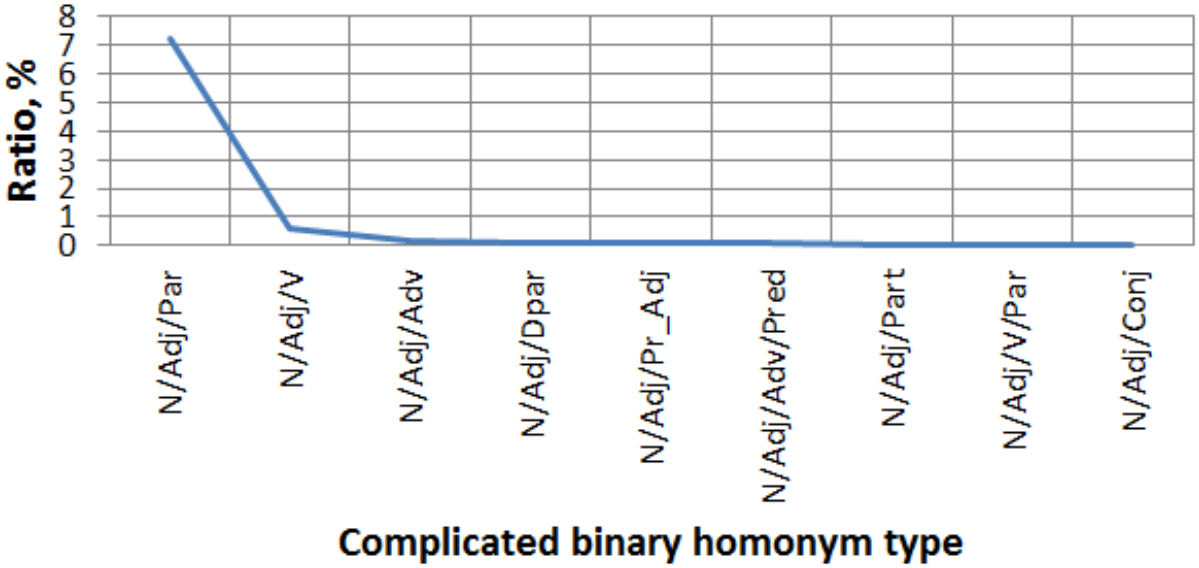


Figure 11. Distribution of complication for binary homonym of N/ADJ/X type

So for describing part of speech aspects of grammatical homonymy in Russian we distinguished four structural parameters:

- *homonymity* parameter (12 strong classes, 5 week classes);
- power of types of grammatical homonymy parameter (126 types);
- binary expression parameter (strong, average, weak values based on different binary characteristics);
- complication of binary parameter (strong, average, weak values based on different binary characteristic).

The detection of distribution of grammatical homonyms implemented by means of different parameters, manifests basic typological features of the Russian language and clarifies intricate transformations of parts of speech in the language use.

Conclusion

Formal and quantitative characterization of comparable structures within and across languages can lead to their complexity ranking. Grammatical homonymy is an important manifestation of structural complexity of a language, and many aspects of it are computable.

This paper deals with problems related to measuring complexity of natural languages on example of determining parameter of describing grammatical homonymy. Grammatical homonymy was examined on linguistic data of the extended version of A. Zaliznyak dictionary, with the help of the software of *Ontointegrator* system.

We distinguished four relevant structural parameters that enable to disclose statistical aspects of part of speech homonymy. Investigation into quantitative aspects of grammatical homonymy implemented by means of different parameters, sheds light on basic typological features of the Russian language and clarifies complicated interconnection of parts of speech in language use.

Grammatical homonymy in Russian can not be reduced merely to part of speech homonymy, so we are planning to expand research area engaging other grammatical categories. The methodology we propose may be used for comparing grammatical homonymy and related phenomena in different languages. In future we are planning to investigate grammatical homonymy in Tatar and to compare it with that of Russian.

Study of complex aspects of grammatical homonymy has important theoretical as well as practical significance, primarily for computer systems for natural language processing, machine translation and machine learning. To refine the methods of machine learning it is necessary to prepare a training collection. In the case of grammatical homonymy, the training collection can and should be built taking into account the complexity and statistical aspects of this phenomenon, relying on the structural model of grammatical homonymy.

Acknowledgement

The work is supported by the Russian Foundation for Basic Research (project # 15-07-09214).

Bibliography

[Bane, 2008] Bane, M. Quantifying and Measuring Morphological Complexity. In Proceedings of the 26th West Coast Conference on Formal Linguistics, 2008. pp. 69-76.

[Becerra-Bonache, 2015] Becerra-Bonache, L., Jimenes-Lopez, M.D. A Grammatical Inference Model for Measuring Language Complexity In Advances in Computational Intelligence. 13th International Work Conference on Artificial Neural Networks, IWANN 2015, Palma de Mallorca, Spain, June 10-12, 2015. Proceedings, Part I, 2015. pp. 4-17.

[Berdichevsky, 2012]. Berdichevsky A. Language Complexity. In Voprosy Yazykoznaniya, Vol. 5, 2012. pp. 101–124. (in Russian)

[Dahl, 2002] Dahl, Ö. The Growth and Maintenance of Linguistic Complexity. John Benjamins Publishing, Amsterdam, 2004.

[Gil, 2008] Gil, D. How Complex are Isolating Languages? In Language Complexity: Typology, Contact, Change. Ed. Miestamo, K. Sinnemäki & F. Karlsson. John Benjamins Publishing, Amsterdam. pp. 109-131.

[Juola, 2008] Juola, P. Assessing Linguistic Complexity. In Language Complexity: Typology, Contact, Change. Ed. Miestamo, K. Sinnemäki and F. Karlsson. John Benjamins Publishing, Amsterdam, 2008. pp. 89 – 108.

[Kusters, 2003] Kusters, W. Linguistic Complexity: The Influence of Social Change on Verbal Inflection. LOT, Netherlands Graduate School of Linguistics, Utrecht, 2003.

[McWhorter, 2001] McWhorter, J. The World's Simplest Grammars are Creole Grammars. In Linguistic Typology. Vol. 5, Issue, 2001: pp. 125-66.

[Miestamo, 2008] Miestamo, M., Sinnemäki, K. and Karlsson, F. (eds). Language Complexity: Typology, Contact, Change. Vol. 94. John Benjamins Publishing, Amsterdam, 2008.

[Nevzorova, 2009] Nevzorova O., Nevzorov V. The Development Support System "OntoIntegrator" for Linguistic Applications. In International Book Series "INFORMATION SCIENCE AND COMPUTING". Number 13. Intelligent Information and Engineering Systems. Supplement to the International Journal "Information Technologies & Knowledge". Vol. 3. ITHEA, Rzeszow-Sofia, 2009. pp. 78-84.

[Newmeyer, 2014] Newmeyer Frederick J. and Preston LaureL B. (ed.) Measuring Grammatical Complexity. Oxford University press, 2014.

[Rastrigin, 1981] Rastrigin L.A. Adaptation of Complex Systems. Methods and Applications. Zinatne, Riga, 1981. (in Russian).

[Zaliznyak, 1987] Zaliznyak A. A. Grammatical dictionary of the Russian Language. Inflection. Russky Yazyk, Moscow, 1987.

Authors' Information

Olga Nevzorova *– Research Institute of Applied Semiotics of Tatarstan Academy of Sciences; Deputy Director. Kazan Federal University. P.O. Box: 420111, Levobulachnaya str., 36a, Kazan, Russia; e-mail: onevzoro@gmail.com*

Major Fields of Scientific Research: Natural language processing, Artificial intelligence

Alfiya Galieva *– Research Institute of Applied Semiotics of Tatarstan Academy of Sciences; Senior researcher. P.O. Box: 420111, Levobulachnaya str., 36a, Kazan, Russia; e-mail: amgalieva@gmail.com*

Major Fields of Scientific Research: Semantics, Grammar of Turkic Languages, Philosophy of Language

Vladimir Nevzorov *– Kazan National Research Technical University named after A.N. Tupolev; Associated Professor the Department of Computer-Aided Design. P.O. Box: 420111, K. Marks str., 10, Kazan, Russia; e-mail: nevzorovvn@gmail.com*

Major Fields of Scientific Research: Natural language processing, Artificial intelligence

USE OF AT-TECHNOLOGY WORKBENCH FOR CONSTRUCTION OF TUTORING INTEGRATED EXPERT SYSTEMS

Rybina G.V., Rybin V.M., Blokhin Yu.M., Sergienko E.S.

National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Moscow, Russian Federation

Email: galina@ailab.mephi.ru

Abstract: *Analyzes the experience of development and use in the educational process MEPhI tutoring integrated expert systems created on the basis of problem-oriented methodology and programming intelligent environment of AT-TECHNOLOGY workbench. The emphasis is on the peculiarities of the implementation of certain tasks of intellectual training, related to the identification of knowledge and skills of students to solve the problem unformalized.*

Keywords: *artificial intelligence, integrated expert systems, problem-oriented methodology, AT-TECHNOLOGY workbench, intelligent software environment, automated planning, tutoring integrated expert systems, intelligent training.*

ITHEA Keywords*: Applications and Expert Systems*

Introduction

Interest in intelligent tutoring systems (ITS) arose at the turn of the XX and XXI centuries, and now they occupy a significant place in a wide scope of intelligent systems issues. . Educational sphere is a good "ground" for the application of artificial intelligence methods and tools, giving rise to a considerable number of approaches and system architectural solutions for intellectualization, individualization and web orientation of learning and training processes . Now, there is an "information explosion" of publications both in Russia and abroad on the subject of ITS. Without claiming to be exhaustive in our review of works in the field of ITS, we will mention only a few papers [Rybina et al, 2016; Smirnova, 2012; Nye, 2015; Rybina, 2011; Bonner et al, 2015], reflecting the results of researches that were conducted in MEPhI and other universities. "Intelligent Systems and Technologies" laboratory at MEPhI's department of Cybernetics has accumulated a lot of experience in the development and use of tutoring integrated expert systems (IES) based on problem-oriented methodology [Rybina, 2008; Rybina, 2014] and powerful modern tools such as AT-TECHNOLOGY workbench [Rybina, 2014; Rybina, 2011; Rybina, 2008] is accumulated in the laboratory "Intelligent Systems and Technologies" department "Cybernetics" MEPhI.

Tutoring IES and web-IES are fully functional, new generation ITS that implement all the basic ITS model (student model, tutoring model, problem domain model, ontology of courses and disciplines, etc.). As well, IES allows solving wide scope of intellectual training tasks, the main ones are [Rybina, 2011; Rybina, 2014; Rybina, 2008]: individual planning of a course / discipline study methodologies; mining solution of educational problems, and intelligent support of decision-making. Process of measuring knowledge level (declarative knowledge of a course/discipline) and detection of skills (procedural knowledge, which shows how this declarative knowledge could be applied in practice) is the basis for all mentioned above tasks. A number of methods for this purpose is proposed. To implement these processes there is a significant number of different methods, according to which the control tests and tasks are developed. For example, in tutoring IES network orientated model of student is formed dynamically on the analysis of answers to questions from special web-tests that are generated with the help of genetic algorithms and the method of estimation is based on calculating the final grade for the whole test. After that the current model of a student knowledge is compared with an ontology of a course/discipline. As a result, one can determine so called "problem areas" in students' knowledge. There are other approaches to identify the level of student's knowledge, as

described, in particular, in [Kehayova et al, 2016; Bonner et al, 2015; Durlach, 2012; Conati, 2012], however, with methodical, algorithmic and technological points of view the implementation of these processes is not particularly difficult. Speaking of ITS with possibility to automatically detect students' abilities to solve problems, there can be difficulties connected with the specifics of a particular course/discipline.

For example, teaching special courses within educational programs like "Applied mathematics and Informatics" and "Software engineering" ("Introduction to intelligent systems", "Expert systems", "Intelligent Information Systems", "Intelligent interactive systems" etc.) is connected with students to do such tasks as [Rybina, 2014]: the ability to build models of the simplest situations in a problem domain based on frames and semantic networks, modeling strategies of forward/backward inference in the ES, construction of linguistic model of business sublanguage and other.

Therefore, to support the construction of tutoring IES on the basis of problem-oriented methodology (AT-TECHNOLOGY workbench) was created and tested in practice in the educational process of MEPhI and other universities special funds that implement "manual" methods of solution of various Non-formalised problems, in particular, is presented in [Gavrilova et al, 2016].

Another important aspect of research and development in the field of ITS is connected with development of tools and technologies for automated support of ITS development. Currently there is no big diversity and innovation of tools and researchers are focused the focus is on reengineering and development of the existing tools [Galeev et al, 2004]. It should be noted that currently there is no standard technology of ITS development, so workbench of general purpose is often used for ITS . For example, [Gribova et al, 2015; Gribova, 2016] The focus of this work is the further development of methods and tools for automated construction of tutoring IES with use of intelligent software environment components.

General characteristics of the components of an intelligent software environment of the AT-TECHNOLOGY workbench

The AT-TECHNOLOGY workbench is a modern tool that supports intelligent software technology for automated construction of IES of different types and levels of difficulty. The conceptual base for the integration of methods of knowledge engineering, ontological engineering, intelligent planning and traditional programming is the concept of "intelligent environments" first introduced in [Rybina, 2008] and studied experimentally in the process of developing a number of applied IES, including tutoring IES [Rybina, 2014; Rybina, 2008; Rybina, 2014; Rybina and Blokhin,2015]. The basic role in the intellectual software environment belongs to the intelligent scheduler, which manages IES and web-IES development projects. Different versions of the scheduler are described in detail in [Rybina, 2008; Rybina, 2014; Rybina and Blokhin,2015] and other works.

Therefore, this work is focused on questions related to the methods of implementation of the above-mentioned tasks of intelligent training with the help of other, equally important components of an intelligent software environment of the AT-TECHNOLOGY workbench. As shown in [Rybina, 2008], the main components of an intelligent software environment used for building and execution of plans for the development of prototypes of applied IES include standard design procedures (SDP) and reusable components (RUC). In accordance with [Rybina, 2008], SDP model for tutoring IES is represented as

$$SDP_T = \langle C_T, L_T, T_T \rangle \quad (1)$$

where C_T is a set of conditions, which ensure SDP invocation; L_T - an execution scenario described with internal SDP actions description language; T_T - a set of parameters initialized by the intelligent planner when SDP is included into an IES prototype development plan. Every RUC, involved in IES prototype development is defined as

$$RUC = \langle N, Arg, F, PINT, FN \rangle \quad (2)$$

N in this model is the name of the component, by which it is registered in the workbench. $Arg = \{Arg_i\}, i = 1...l$ - set of arguments containing current project database subtree serving as input parameters for the functions

from the set. $F = \{F_i\}, i = 1...s$ - a variety of methods (RUC interfaces) for this component at the implementation level. $PINT$ - a set of other kinds of RUC interfaces, used by the methods of the RUC. $FN = \{FNi\}, i = 1...v$ - set of functions names performed by this RUC. The main algorithm element used during development plan generation process of the IES prototype is SDP. By SDP we mean a set of elementary instructions (steps) which are traditionally executed by a knowledge engineer at every development lifecycle stage. The intelligent planner of the AT-TECHNOLOGY workbench has knowledge about all available SDPs, and based on this knowledge it forms a set of tasks for any IES prototype development (accordingly to a current lifecycle stage). Then, basing on special requirements specified at the system requirements analysis stage, the planner decomposes the plan into smaller tasks (subtasks). All the workbench SDPs are classified in the following manner: task type independent SDPs

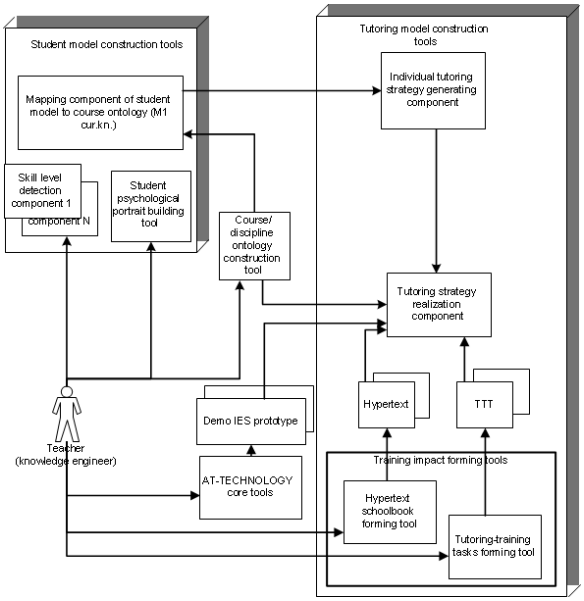


Figure 1: Execution scheme of the SDP "Tutoring web-IES construction" in DesignTime mode for a teacher (or a knowledge engineer)

(for example, "knowledge acquisition from database"), task type dependent SDPs (for example forming tutoring IES components), SDPs related with RUC, i.e. procedures, that contain knowledge about RUC lifecycle from its configuring up to including it into the IES prototype model. SDPs of the last type also contain knowledge about problems solved with this RUC and its necessary configurations. The general architecture of the AT-TECHNOLOGY workbench is built in such manner, that all functionality is distributed between the components registered in the workbench and acting under intelligent development environment. In other words, these components are reusable components of the workbench, and they are developed in accordance with some workbench rules [Rybina, 2008].

There are two different types of RUC used in the current basic AT-TECHNOLOGY workbench version - procedural and informational components. In the first one the components provide capabilities for execution either actions with non-typical results, i.e. results that are not stored in some special storage (repository) as the results of previous developments, or actions, that require user interaction (for example, editing the ER-scheme or viewing the expert interviewing protocol). In the second one the components provide capabilities for executing actions which result in the information that has been collected earlier and is stored in the repository (knowledge, data, schemes, structures etc.) with further copying of this information into the current project and preprocessing if needed (i.e. copying of created earlier ER-diagram or typical diagram analysis). Special storages (repositories) are used for RUC of the second type. They collect different types of data which is used in further development processes.

In the basic AT-TECHNOLOGY workbench many SDPs of the first and second types are implemented and used, in particular: the SDP for combined knowledge acquisition, the SDP for database designing, the SDP for configuring IES prototype components, the SDP for creating hyper-text tutorials etc. There are SDPs related to distributed knowledge acquisition from different knowledge sources, dynamic IES development SDP and the most complicated

SDP for tutoring IES construction [Rybina, 2008] in the experimental stage. The difficulties of the tutoring IES development technology are caused by supporting two different work modes - DesignTime, oriented to work with teachers (course/discipline ontology creating processes, different typed training impacts creating, etc.) and Runtime, for working with students (current student model building processes, including psychological model, etc.). The execution scheme of SDP "Tutoring web-IES construction" in the DesignTime mode is presented in Fig. 1, and in the RunTime mode - in Fig.2.

As shown in Fig. 2., in the RunTime mode the following AT-TECHNOLOGY workbench instruments are used: training impact building tools (hypertext schoolbook, tutoring-training tasks), basic core workbench tools for IES prototype construction, student psychological model builder tool, tools for course/discipline ontology building, individual tutoring strategy former, tutoring strategy realization component, different skills level detection components, component for mapping a current student model to course/discipline ontology. As shown in the scheme of Fig.2. the student

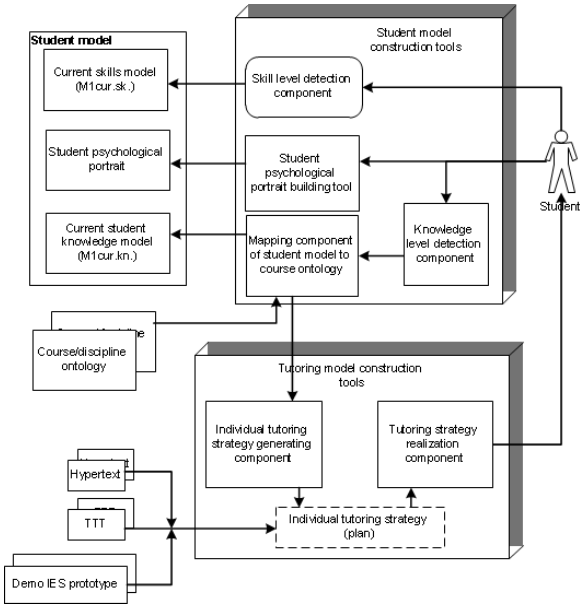


Figure 2: Execution scheme of the SDP "Tutoring web-IES construction" in the RunTime mode for students

model construction (student model current knowledge, student model current skills, psychological model components) is directly connected to the tools for construction and realization of the tutoring model, as well as to the component for mapping a current student model to course/discipline ontology. The mapping component is also connected to the individual tutoring strategy forming component. The peculiarity of a tutoring web-IES developed with AT-TECHNOLOGY workbench is a presence of some components of skill level detection and of the component for building a student's psychological model (as an aggregate of personal characteristics which are collected as a results of psychological testing).

Consider this SDP in context of topicality of intelligent development technology usage for tutoring IES. The schemes of the SDP "Tutoring web-IES construction" shown in Fig.1. and Fig.2. clearly show a big amount of repeating routine operations which must be performed by a knowledge engineer during the designing lifecycle stage (DesignTime mode) and maintenance (RunTime mode) of the tutoring web-IES for certain courses/disciplines.

The most difficult and complicated stage is the construction of the "Training with IES" training impact, which includes a comprehensive problem of the applied IES development for a certain problem. This IES is developed using AT-TECHNOLOGY workbench core. For example, almost all the courses of the "Intelligent systems and technologies" specialization require some knowledge of engineering methods. These methods are presented as non-formalized tasks and non-formalized methods such as "System analysis of the problem domain about applicability of the ES technology", "Choosing knowledge representation formalism", "Choosing development tools" and other tasks requiring expert knowledge [Rybina, 2008]. The aggregate of the listed non-formalized tasks and their logical

relations are the base for the problem domain construction. The problem domain is constructed with the knowledge representation language used in the AT-TECHNOLOGY workbench.

Features of realization of some intelligent training tasks based on the use of operational and informational RUC

In accordance with problem-oriented methodology for constructing IES [Rybina, 2011], one of the important components of a generalized model of a typical tutoring problem is the network student model, the construction and renovation of which is carried out dynamically by implementing control measures, provided by the curriculum of each course / discipline. For these purposes, as a part of the subsystem of construction of tutoring IES / web-IES, which works in both (basic and web version) of AT-TECHNOLOGY workbench, there are special tools for building student model.

As described above, the set of SDP and RUC are components of intellectual technological knowledge base [Rybina, 2011; Rybina, 2008] providing the intellectualization of the creation and operation of a wide class of IES, including tutoring IES. Detailed description of the intelligent software environment model and the means of its implementation are given in [Rybina, 2014; Rybina and Blokhin,2015]. Specifications of operational and informational RUC's that copes with major issues of intellectual tutoring are presented in this work. Brief description of these RUC's is given below.

01 Individual planning of studying methods of a course

Main operational RUCs for this task are the means of building the ontology of a course/discipline [Rybina et al, 2012]. Also it uses about ten informational RUC's, associated with fragments of hypertext electronic textbooks (HT-books) for specific courses / disciplines, and several informational RUC's for building a generalized ontology "Intelligent systems and technologies" and ontology "Automation of physical installations and scientific research" (Department of Automation [Rybin, 2011]). In general, the current RUC's for construction of a student model are the following:

1. Operational RUC "Component that identifies a student's level of knowledge" (and several informational RUC's that describes test problems for various fragments of a course/discipline ontology);
2. Four operating RUC's associated with evaluating the level of skills of a student include: component of detection of student's skills to simulate the forward / backward inference, component of detection of skills to build components of a linguistic model of business prose sublanguage, component of detection of skills to simulate the simplest situation in problem domain using frames and components of detection of skills to simulate the simplest situation in problem domain using semantic networks;
3. For ontology's, "Automation of physical installations and of scientific research" [Rybin, 2011] is used the operational RUC associated with the detection of learners' abilities to develop automatic control systems (ACS) over physical units ("Physical Component ACS units").
4. Two operational RUC's - "Psychological test generator" and "Component for student personal characteristics detection". The process of generation of psychological tests is carried out using informational RUC's containing fragments copyright psychological tests aimed at identifying the set of personal characteristics of students.

It should be noted that the component for displaying current student model, compared with ontology of a course / discipline, and designed as an operational RUC. It allows to reveal "problem areas" of a student. That helps to construct the individual plan (strategy) of tutoring. Figure 2 shows the architecture of tools for building a learning model and for automatic generation of an individual learning plan, that uses operational RUC "Component of forming tutoring plans (strategies)", and a special RUC "Component of managing the application of tutoring impact ".

Each training strategy includes a specific sequence of tutoring impacts such as: reading of a hypertext book; solution of several types of training problems ("Building relationships between elements of the graphical representation,"

"Organizing graphics", "Enter a numeric value for the interval", "graphic analysis", "The mapping and sequencing of the blocks", "Formation of the answer by selecting its components from the proposed list "," Marking the correction of the text"," Filling the gaps in the text ","Setting correspondences between blocks "," Enter the answer to the open question"); implementation of tutoring impact "Training with IES"; explanation of the obtained results; tips; localization of errors made; control of the correct solutions, etc. Any tutoring strategy is characterized by a specific set of procedures and application of tutoring impacts, the content of which is determined by the degree of destabilization of the problem, depending on the level of knowledge and skills of a student and his or her psychological portrait. The process of formation and implementation of all relevant tutoring impacts is supported by special operational and informational RUC's.

02 Intelligent analysis of tutoring problems solutions

To identify the skills and abilities of students to solve tutoring non-formalized problems from six courses/disciplines represented in a generalized ontology "Intelligent Systems and Technologies" [Rybina, 2014] a simulation of student's reasoning for solving four types of learning tasks was used: modeling strategies of forward / backward inference, simulation of simple situations of problem domain using frames and semantic networks, building the components of a linguistic model of business prose sublanguage. Let's briefly comment on operational RUC's that support the above tasks.

1. Operational RUC "Component of detection of student's skills to simulate the forward / backward inference" and several informational RUC's (fragments of knowledge bases) are designed to identify the learner's skills to simulate the forward / backward inference (courses "Introduction to Intelligent Systems", "Expert System", "Intelligent information Systems" etc.). Students go through the following steps: create DBs, consisting of production rules; input initial facts for direct inference; model a strategy of forward inference; input facts and goals for backward inference; model strategy of backward inference. Students skills are evaluated with a simple solver, performing standard inference, and then this inference is compared (using special heuristics) with the students solution.

2. Operational RUC "Component of detection of skills to simulate the simplest situation in the problem domain using frames" and several informational RUCs (fragments of prototype frames, in FRL language of knowledge representation [Rybina, 2014]) provide the functionality declared in the course "Introduction to Intelligent Systems", "Expert Systems", "Intelligent information Systems". Students create prototype frames defined by a tutor [12], then by comparison with the reference frames the level of skills of a students is detected. A complete history of student actions is saved and can be used to reproduce student's logic of reasoning.

3. Operational RUC "Component of detection of skills to simulate the simplest situation in problem domain using semantic networks" and several informational RUC's (fragments of semantic networks) provide functionality declared in the courses "Introduction to Intelligent Systems", "Expert systems" and "Intelligent Information Systems". Students construct a fragment of a semantic network for a given problem domain, and then on the basis of comparison with reference fragments of the semantic network the level of their skills is defined with the help of expert techniques.

4. Operational RUC "Component of detection of skills to build components of a linguistic model of business prose sublanguage" and several informational RUC's (dictionaries, fragments of business texts, etc.) provide the functionality declared in the course "Intelligent interactive systems". Students do control tasks of creating lexical, syntactic and semantic components of a linguistic model for a business prose text sublanguage, and then the level of their skills is defined with the help of a special expert techniques. To identify the skills/abilities of students to solve both formal and Non-formalised-problems in the ontology "Automation of physical installations and scientific research" the operating RUC "Design of automation of physical installations", which provides the following functionality in the appropriate course/discipline [Rybin, 2011]: development of block diagrams of ACS; calculation of stability of ACS; the choice of ACS elements.

03 Intelligent Decision Support

It is important to note that in the development of tutoring impacts such as "Training with IES" for different formalized courses/disciplines the most important task is building of problem domain models (including those based on knowledge, containing certain types of NE-factors [Rybina, 2008]). Another important task is implementation of "consultation with IES" mode, in which there are scenarios of dialogues with the student. In this dialogues a considerable attention is given to explanations, tips and / or verification of the next stage of solving the problem, etc. Here we could apply multiple operational RUC's from the basic AT-TECHNOLOGY workbench (communication subsystem, universal AT-solver, an explanation subsystem, etc.), as the development of tutoring impact is a task of creating a complete IES. Informational RUC's are also used (knowledge based fragments from previously created teaching operations "Training with IES", fragments of user dialogue scenarios in "Consultation with IES" mode, etc.) and operational RUC "Explanation component" provides assistance at every stage of the solution of educational problems particularly, gives hints of the next stage, gives explanations like "how" and "why" as well as makes at visualization of inference.

Conclusion

Currently, we are doing a pilot software study, re-engineering and further development of all components of intelligent technologies of tutoring IES construction. In addition, we are working on implementation of Non-formalized techniques for solving tutoring problems in other courses of various ontologie's (in particular the "Dynamic intelligent systems", etc.)

Acknowledgements

The work was done with the Russian Foundation for Basic Research support (project 15-01-04696)

Bibliography

[Rybina et al, 2016] G. V. Rybina and V. M. Rybin and E. S. Sergienko. Some features of development and using of tutoring integrated expert systems in educational process mephi. In: Proceedings of the VI International Scientific and Technical Conference, 18-20 February, Minsk. 2016.

[Rybina, 2011] G. V. Rybina. Intelligent tutoring systems based on integrated expert systems: the experience of the development and use. IJ Information-measuring and operating systems, No.: 10, pp. 4–16. 2011

[Rybina, 2014] G. V. Rybina. Intelligent systems: from A to Z. Monography series in 3 books. Vol. 1. Knowledge-based systems. Integrated expert systems. Nauchtehlitizdat, 2014, Moscow. p.224,

[Rybina, 2008] G. V. Rybina, Theory and technology of construction of integrated expert systems. Monography. Nauchtehlitizdat, 2008, Moscow. p.482,

[Rybina, 2014] G. V. Rybina. Fundamentals of building intelligent systems. Tutorial. Finance and Statistics, 2014, Moscow. p. 432,

[Rybin, 2011] V. M. Rybin. Intelligent control based on dynamic integrated expert systems. IJ Information-measuring and operating systems, No.: 6, pp. 16–19. 2011.

[Rybina et al, 2013] G. V. Rybina and Y. M. Blokhin and M. G. Ivashenko. Some aspects of intelligent technology for integrated expert system construction. IJ Devices and Systems. Control, monitoring, diagnostics, No.: 4, pp. 27–36. 2013.

[Rybina and Blokhin,2015] G. V. Rybina and Y. M. Blokhin. Modern automated planning methods and tools and their use for control of process of integrated expert systems construction. IJ Artificial intelligence and decision making, No.: 1, pp. 75–93, 2015.

[Rybina et al, 2012] G. V. Rybina and M. V. Yusova and E. V. Churdalev. Ontologies in the training of expert integrated systems. IJ Information-measuring and operating systems, No.: 8, pp. 13–20, 2012.

[Rybina, 2011] G. V. Rybina. Instrument complex AT-TECHNOLOGY to support building integrated expert systems: general characteristics and development prospects. IJ Devices and Systems. Control, monitoring, diagnostics. No.:11, pp. 17–40, 2011.

[Smirnova, 2012] N. V. Smirnova and A. Y. Schwartz. Motivational and volitional component of the student model in intelligent tracking systems. Part 1. IJ Artificial intelligence and decision making, No.:1, pp. 65–80, 2012.

[Nye, 2015] B.D. Nye. Intelligent tutoring systems by and for the developing world: A review of trends and approaches for educational technology in a global context. International Journal of Artificial Intelligence in Education, No.: 25. pp. 177–203. 2015.

[Gavrilova et al, 2016] T. A. Gavrilova and D. V. Kudryavtsev and D. I. Muromtsev. Knowledge Engineering. Models and methods: A Textbook. // Lan, Saint Petersburg, 2016. p. 324,

[Kehayova et al, 2016] I. Kehayova and P. Malinov and V. Valkanov and E. Doychev. Architecture of a Module for Analyzing Electronic Test Result. In: Proceedings of 2016 IEEE 8th International Conference on Intelligent Systems (IEEE IS'16). 2016, Sep. 4-6.

[Bonner et al, 2015] D. Bonner and J. Walton and M.C. Dorneich and S.B. Gilbert and E. Winer and R.A. Sottilare. The development of a testbed to assess an intelligent tutoring system for teams. In: Workshops at the 17th International Conference on Artificial Intelligence in Education, AIED-WS 2015; CEUR Workshop Proceedings, 2015, Sep. 4-6.

[Durlach, 2012] P.J. Durlach and A.M. Lesgold. Adaptive technologies for training and education. Cambridge University Press, London 2012, p. 360

[Conati, 2012] C. Conati, Student modeling and intelligent tutoring beyond coached problem solving. IJ Adaptive Technologies for Training and Education, No.:1, pp. 96–116, 2012

[Galeev et al, 2004] I. Galeev and L. Tararina and O. Kolosov. Adaptation on the basis of the skills overlay model. In: Procedings of the 4th IEEE International Conference on Advanced Learning Tecknologies (ICALT 2004), 2004,Sept.

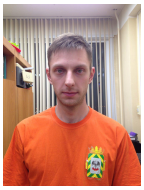
[Gribova et al, 2015] V.V. Gribova and A.S. Kleschev and D.A. Krylov and F.M. Moscalenko. The basic technology development of intelligent services on cloud platform IACPaaS. Part 1. The development of a knowledge base and a solver of problems, IJ Software engineering, No.:12, pp. 3–11, 2015. âĂŞ Smolensk: Universum, 2016. Đć. 3. Đą.171-179

[Gribova, 2016] V.V. Gribova and G.E. Ostrovskiy, Intelligent learning environment for the diagnosis of acute and chronic diseases, Proceedings of the XV national conference on artificial intelligence with international participation KII-2016), 2016, Oct. 1-4,

Authors' Information

Galina V. Rybina *- Doctor of Technical Science, Professor cybernetics department of National Research Nuclear University MEPhI (Moscow Engineering Physics Institute). RF President education award winner. Full member Academy of Natural Sciences. Accents on intelligent systems and technologies, static, dynamic and integrated expert systems, intelligent dialogue systems, multi-agents systems, workbenches; e-mail:galina@ailab.mephi.ru*

Victor M. Rybin *- Doctor of Technical Science, Professor department of Automation of National Research Nuclear University MEPhI (Moscow Engineering Physics Institute). Full member Academy of Natural Sciences. Accents on automation and electronics, electro physical complex, automatic control system, intelligent control systems, dynamic intelligent systems; e-mail:vmrybin@yandex.ru*



Yuri M. Blohin *- assistant department of cybernetics of National Research Nuclear University MEPhI (Moscow Engineering Physics Institute). e-mail: ultrablox@gmail.com*

Elena S. Sergienko *- graduate student department of cybernetics of National Research Nuclear University MEPhI (Moscow Engineering Physics Institute). e-mail: deav@inbox.ru*

RESEARCH ON THE PROPERTY "AVALANCHE EFFECT" IN IDA CRYPTOGRAPHIC ALGORITHM

Ivan Ivanov, Stella Vetova, Krassimira Ivanova, Neli Maneva

Abstract: *The following paper presents some conducted extensive research on the cryptographic algorithm IDA, concerning one of the basic properties of the block algorithms "avalanche effect". The subject of the research are two different open texts, differing only by one bit and one key, as well as two keys differing only by one bit and one open text.*

Keywords: *cryptography, cryptographic algorithm, avalanche effect, S matrix, IDA algorithm*

ITHEA Classification Keywords: *E.3 Data Encryption – cryptosystems; F. Theory of Computation: F.2 Analysis of Algorithms and Problem Complexity; K. Computing Milieux: K.7 The Computing Profession: K.7.3 Testing, Certification, and Licensing*

Introduction

The purpose of the present paper is the exploration of the IDA algorithm property "avalanche effect" [Ivanov et al., 2014] in the following main tasks: (1) introduction of the property "avalanche effect"; (2) research on the IDA algorithm property "avalanche effect"; (3) results analysis.

The property "avalanche effect"

High result sensibility for initial data alteration is a desirable property for most of the encryption algorithms. According to its essence, any small alteration of the clear text or key should lead to a significant alteration in the ciphertext [Stallings, 2013; Schneier, 2013]. In particular, alteration of any single bit of the clear text or key should lead to the value alteration of great amount of the ciphertext bits [Sokolov & Shangin, 2002]. Even if the alteration in the ciphertext is small, it may cause a significant reduction of the set of keys or the field of the clear text.

Research on the IDA algorithm property "avalanche effect"

To research the IDA algorithm property "avalanche effect", two different clear texts will be encrypted. In this case, both texts differ by only one bit:

P_1 = 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

P_2 = 10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

The key is the same:

K = 11101010 11101110 11110000 11100101 11101010 11110010 11101110 11110000 00100000 11101101 11100000 00100000 11110010 11100101 11101011 11100101 11101010 11101110 11101100 11110011 11101101 11101000 11101010 11100000 11110110 11101000 11101110 11101101 11101101 11101000 11110010 11100101

For the clear text encryption, the research is similar:

P = 11110010 11100101 11101011 11100101 11110100 11101110 11101101 11101000, and two keys which differ by one bit:

K_1 = 11101010 11101110 11110000 11100101 11101010 11110010 11101110 11110000 00100000 11101101 11100000 00100000 11110010 11100101 11101011 11100101 11101010 11101110 11101100 11110011 11101101 11101000 11101010 11100000 11110110 11101000 11101110 11101101 11101101 11101000 11110010 11100101

K_2 = 01101010 11101110 11110000 11100101 11101010 11110010 11101110 11110000 00100000 11101101 11100000 00100000 11110010 11100101 11101011 11100101 11101010 11101110 11101100 11110011 11101101 11101000 11101010 11100000 11110110 11101000 11101110 11101101 11101101 11101000 11110010 11100101

The research results are tabled in Table 1.

Table 1. Research the IDA algorithm property "avalanche effect"

Plain text alteration		**Key alteration**	
Loop	**Difference (bits)**	**Loop**	**Difference (bits)**
0	5	0	4
1	14	1	12
2	25	2	18
3	37	3	30
4	39	4	35
5	35	5	31
6	32	6	30
7	31	7	32
8	29	8	32
9	41	9	38

Plain text alteration		Key alteration	
Loop	**Difference (bits)**	**Loop**	**Difference (bits)**
10	39	10	40
11	32	11	33
12	30	12	29
13	30	13	26
14	29	14	30
15	36	15	35

Figure 1 graphically represents the results of the Table 1.

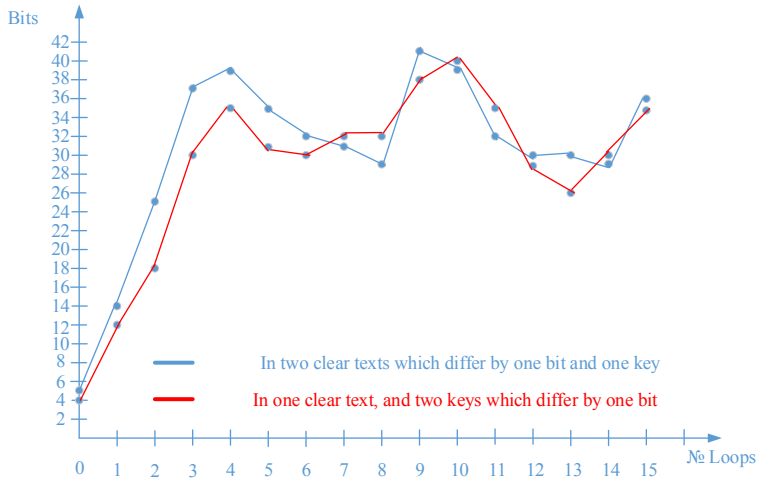


Figure 1. Research results on the IDA algorithm property "avalanche effect"

As can be seen from Figure 1 and Table 1, IDA algorithm has strong avalanche effect. Yet, it is seen that after the third encryption loop there is a difference of 37 bits. At the end of the encryption process, there is a difference of 36 bits.

On the analogy of the first case, in the clear data encryption using keys which differ by one bit (Figure 1 and Table 1), the avalanche effect is strong too. It is seen that after the third encryption loop, there is a difference of 30 bits. At the end of the encryption process, there is a difference of 35 bits.

To compare the results, the DES algorithm is put to test in the conditions described earlier. Table 2 and Figure 2 depict the obtained results.

As Figure 2 and Table 2 show, the DES algorithm demonstrates strong avalanche effect too. It is also seen that after the third loop of the encryption process, a difference of 35 bits occurs. At its end, there is a difference of 34 bits.

Similar to the first case, in the clear data encryption using two bits which differ by one bit (Figure 2 and Table 2) the avalanche effect is strong too. It is clearly seen that after the third encryption loop, there is a difference of 28 bits. At the end of the encryption process, the difference reaches 35 bits.

Table 2. Research the DES algorithm property "avalanche effect"

Plain text alteration		**Key alteration**	
Loop	**Difference (bits)**	**Loop**	**Difference (bits)**
0	1	0	0
1	6	1	2
2	21	2	14
3	35	3	28
4	39	4	32
5	34	5	30
6	32	6	32
7	31	7	35
8	29	8	34
9	42	9	40
10	44	10	38
11	32	11	31
12	30	12	33
13	30	13	28
14	26	14	26
15	34	15	35

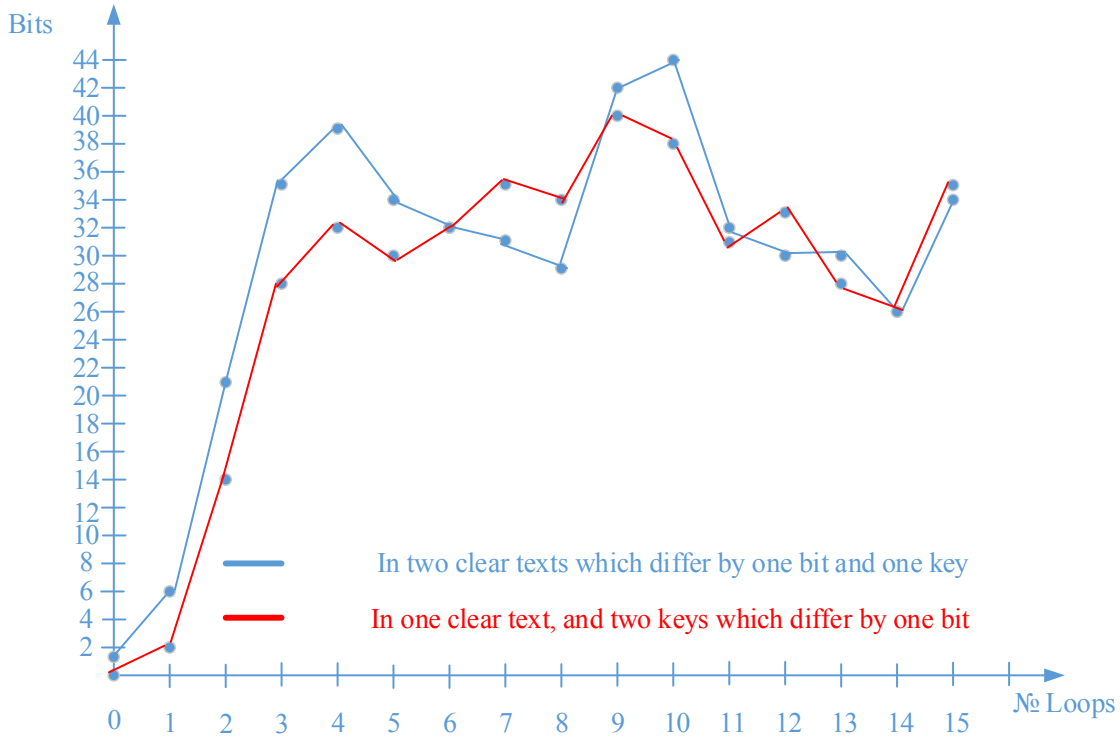


Figure 2. Research results on the DES algorithm property "avalanche effect"

Conclusion

As a result of the performed research work, there are three obtained results:

1. In the IDA algorithm in two clear texts which differ by one bit and one key, after the third encryption loop there is a mean difference of 35 bits from the total 64 bits for the rest twelve loops;
2. In the IDA algorithm in one clear text, and two keys which differ by one bit after the third loop of the encryption process, there is a mean difference of 33 bits from the total 64 bits for the rest twelve loops;
3. The IDA algorithm possesses a better avalanche effect compared to the DES algorithm (mean difference of three bits).

Acknowledgements

The paper is published with partial financial support from the "Scientific Research Fund" of University of Telecommunications and Posts, Sofia, Bulgaria, by the research project "Methods for development and estimation of cipher functions in block encryption algorithms".

Bibliography

[Ivanov et al., 2014] Ivanov I, Arnaudov R, Dikov D, Stanchev G, Patent application 111513: Method for increasing data security in storage and during information transmission in special purpose telemetry systems, Bulgarian patent office, Official Bulletin, Issue 12, pp.14, Dec 2014.

[Katz & Lindell, 2014] Katz J., Lindell Y. Introduction to Modern Cryptography, Second Edition (Chapman & Hall/CRC Cryptography and Network Security Series), CRC Press, 2014.

[Schneier, 2013] Schneier B., Appliet Cryptography Protocols, Algorithms, and Source Code in C, Wiley, 2013.

[Sokolov & Shangin, 2002] Sokolov V., Shangin F. Information protection distribured corporate networks and systems. DMK Press, Moskva, 2002.

[Stallings, 2013] Stallings W. Cryptography and Network Security: Principles and Practice (6th Edition), Hardcover, 2013.

Authors' Information

Ivan Ivanov *– Assist. Prof. PhD; University of Telecommunications and Posts, Sofia, Bulgaria; e-mail: i.ivanov@utp.bg;*

Major Fields of Scientific Research: Information and Network Security, Cryptographic Methods and Algorithms, Cyber security.

Stella Vetova *– Scientific Researcher, e-mail: vetova.bas@gmail.com*

Major Fields of Scientific Research: Databases and security, Artificial Intelligence, Computer networks.

Krassimira Ivanova *- Assoc. prof. Dr.; University of Telecommunications and Posts, Sofia, Bulgaria; Institute of Mathematics and Informatics, BAS, Bulgaria; e-mail: krasy78@mail.bg;*

Major Fields of Scientific Research: Software Engineering, Business Informatics, Data Mining, Multidimensional multi-layer data structures in self-structured systems

Neli Maneva *– student; University of Telecommunications and Posts, Sofia, Bulgaria; e-mail: i.ivanov@utp.bg;*

Major Fields of Scientific Research: Information and Network Security, Computer networks and protocols.

INDUCTIVE MODELING METHOD GMDH IN THE PROBLEMS OF DATA MINING

Yuriy Zaychenko, Galib Hamidov

Abstract: *The problem of constructing unknown dependencies (laws) in huge data warehouses is considered. For its solution inductive modeling method- so called Group Method of Data Handling (GMDH) is suggested. This method enables to construct automatically optimal models of variables based on experimental data stored in data warehouses. Unlike other modeling methods GMDH enables to find out the structure of the unknown model and solves the problem of not parametric, but structural identification. Additionally for finding unknown laws in incomplete and unreliable data under uncertainty fuzzy GMDH is suggested enabling to construct fuzzy models. The experimental investigations of the suggested methods for models identification in Data Mining problems are presented and the obtained results discussed.*

Keywords: *Data Mining, GMDH, fuzzy, model identification.*

ITHEA Keywords*: I. Computing methodologies; I 2. Artificial intelligence, I 6.5. Model development*

Introduction

Last year's problems of Data Mining in data bases (DB) have become very crucial in IT-applications. Especially it refers to big DB, so-called data warehouses where mountains of raw data are accumulated and hidden laws in these data are to be detected and corresponding models to be constructed [Barsegyan, 2008; Duke, 2001].

Previously several classes of methods were developed for finding unknown dependencies in data, in particularly statistical methods: ARMA, Logit and Probit models, ARCH and GARCH methods and neural networks. But they have drawbacks: statistical methods solve only problems of parametric identification and don't solve structural identification while neural networks allow to determine model structure but in an implicit form. The model structure is hidden in neural weights and its analytical form is unavailable.

Therefore the development of methods for structural models identification constitute important problem in DM. The main goal of this paper is development and investigation of methods for constructing models in data accumulated in data warehouses. For this goal the method of inductive modeling- Group Method of Data Handling (GMDH) is suggested and investigated [Ivakhnenko, 1985].

For finding unknown laws in data under uncertainty new version of GMDH – Fuzzy GMDH is suggested and explored [Zaychenko, 2003; Zaychenko, 2006]. Fuzzy GMDH enables to operate with incomplete or indefinite initial data and constructs fuzzy models whose coefficients are fuzzy.

The significant property of GMDH is that it may operate with high dimensional data (with many variables) and so-called "short samples" when the number of model coefficients m is greater than sample size N. This is achieved due to specificity of GMDG algorithm as at each step of it a set of so-called partial models are constructed consisting only of two variables instead of n initial input variables like other modeling methods. This enables to cut substantially the dimension of model and decrease the time for its construction. This advantage rises with the increase of model complexity: the greater is model dimension (number of variables), the greater is cut in computational time for its construction as compared with conventional modeling methods.

1. Problem Formulation

Consider the problem of model construction. A set of initial data is given, inclusive input variables $\{X(1), X(2), ..., X(N)\}$ and output variables $\{Y(1), Y(2), ..., Y(N)\}$, where $X = [x_1, x_2, ..., x_n]$ *is* -n-tuple vector, N is a number of observations .

The task is to synthesize an adequate forecasting model $Y = F(x_1, x_2, ..., x_n)$, and besides, the obtained model should have the minimal complexity. In particularly, while solving forecasting problem as an output variable Y a forecasting model is used $X(N+K) = f(X(1), ..., X(N))$, where K is a value of a forecasting interval.

The constructed model should be adequate according to the initial set of data, and should have the least complexity (Figure 1).

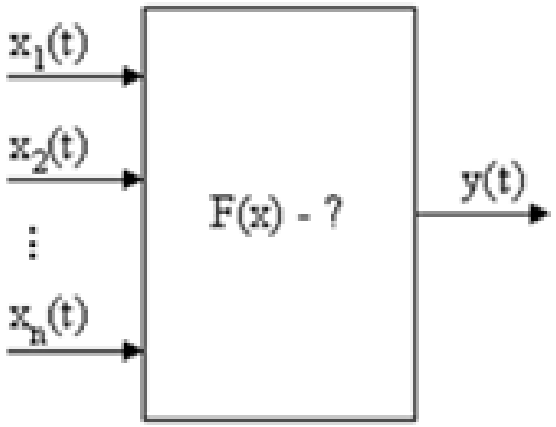


Figure 1. Graphical representation of the problem

The distinguishing features of the problem are the following:

1. Form of functional dependence is unknown and only model class is determined, for example, polynomial of any degree or Fourier time series.
2. Short data samples;
3. Time series $x_i(t)$ in general case is non- stationary.

In this case the application of conventional methods of statistical analysis (e.g. regression analysis) is impossible and it's necessary to utilize methods based on computational intelligence (CI). To this class belongs Group Method of Data Handling (GMDH) developed by acad. A. Ivakhnenko [Ivakhnenko, 1985] and extended by his colleges. GMDH is a method of inductive modeling. The method inherits ideas of biological evolution and its mechanisms:

1. Crossing-over of parents and offspring generation;
2. Selection of the best offsprings.

GMDH method belongs to self-organizing methods and allows to discover internal hidden laws in the appropriate object area.

The advantages of GMDH algorithms are the possibility of constructing optimal models with a small number of observations and unknown dynamics among variables. This method doesn't demand to know the model structure a priori, the model is constructed by algorithm itself in the process of its run.

The basic principles of GMDH

Let's remind the fundamental principles of GMDH [3, 4-6]. The full interconnection between input $X(i)$ and output $Y(i)$ in the class of polynomial models may be presented by so-called generalized polynomial of Kolmogorov- Gabor:

$$Y = a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{j=1}^{n} \sum_{i \le j} a_{ij} x_i x_j \sum_{i=1}^{n} \sum_{j \le i} \sum_{k \le j} a_{ijk} x_i x_j x_k + \ldots \quad (1)$$

where all the coefficients $\bar{a}_0, \bar{a}_i, \bar{a}_{ij}$, are unknown.

While constructing model (search coefficients values) as a criterion of adequacy the so-called regularity criterion (mean squared error- MSE) is used

$$\overline{\varepsilon^2} = \frac{1}{N} \cdot \sum_{i=1}^{N} (y_i - f(X_i))^2 \qquad (2)$$

where *N* is a sample size (number of observations).

It' s demanded to find minimum $\overline{\varepsilon}^2$.

GMDH method is based on the following principles [Ivakhnenko, 1985, Zaychenko, 2003].

The principle of multiplicity of models. There is a great number of models providing zero error on a given sample . It's enough simply to raise the degree of the polynomial model. If N nodes of interpolation are available, then it's possible to construct the family of models each of which gives zero error on experimental points $\overline{\varepsilon}^2 = 0$.

The principle of self-organization. Denote S as model complexity. The value of an error depends on the complexity of a model. As the the level of complexity S grows the error first drops, attains minimum value and then begins to rise (see Fig. 2).

We need to find such level of complexity for which the error would be minimal. In addition if to take into account the action of noise we may make the following conclusions concerning ε:

1. With the increase of noise the optimal complexity $s_0 = \arg\min \overline{\varepsilon}^2$ shifts to the left;
2. With the increase of noise level the value of optimal criterion min $\overline{\varepsilon}^2(s)$ grows.

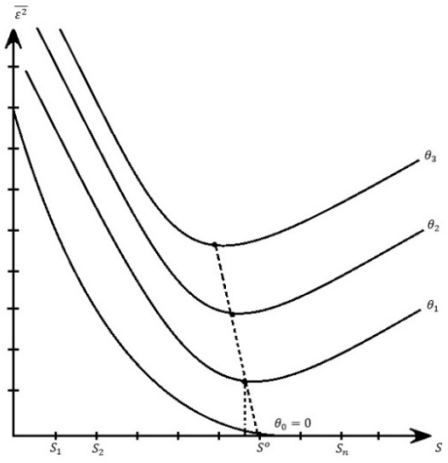


Figure 2. Dependence of criterion $\overline{\varepsilon}^2$ on model complexity S

Theorem of incompleteness by Geodel: In any formal logical system there are some statements which cannot be proved or refuted using the given system of axioms and staying in the margins of this system. That to prove or refute such statement one need go out this system and use some external information (meta information) which is called "external complement". In our case as external information stands additional sample of data which wasn't used for the finding unknown coefficients of the model.

So one way to overcome incompleteness of sample is to use **principle of external complement** which means that the whole sample should be divided into two parts – training subsample and test subsample. The search of optimal model is performed in such a way:

- At the training sample N_{train} the estimates $\bar{a}_0, \bar{a}_i, \bar{a}_{ij}$, are determined;
- At the test sample N_{test} the best model is selected.

The ideas of computational method GMDH

For each pair of inputs x_i and x_j so-called partial descriptions are being built (all in all C_n^2) of the form:

$$\bar{Y}_s = \phi(x_i, x_j) = a_0 + a_i x_i + a_j x_j \ , \ s = 1..C_n^2 \ \text{(linear)};$$
$$\text{or } \bar{Y}_s = \phi(x_i, x_j) = a_0 + x_i + a_j x_j + a_{ii} x_i^2 + a_{ij} x_i x_j + a_{jj} x_j^2 \ , \ s = 1..C_n^2 \ \text{(quadratic)}. \quad (3)$$

1. Determine the coefficients of these model using LSM (least square method) at the training sample (i.e. find estimates $\bar{a}_0, \bar{a}_1, ..., \bar{a}_j, ..., \bar{a}_N, \bar{a}_{11}, ..., \bar{a}_{ij}, ..., \bar{a}_{NN}$.
2. Further at the test sample for each of these models calculate the value of regularity criterion :

$$\bar{\delta}_s^2 = \frac{1}{N_{test}} \cdot \sum_{i=1}^{N_{testx}} [Y(k) - \bar{Y}_s(k)]^2 \quad (4)$$

(where $Y(k)$ is real output value of the k-th point of test; $\bar{Y}_s(k)$ is a value of this criterion on k-th point obtained by model, N_{test} is a number of points at the test sample);

as alternate criterion "unbiasedness" criterion may be used:

$$N_{ub} = \frac{1}{N_1 + N_2} \sum_{k=1}^{N} \left(y_k^* - y_k^{**} \right)^2 \quad (5)$$

where the sample is also divided in two parts N_1 and N_2, y_k^{**} are outputs of the model built on the subsample N_1, y_k^{**} are outputs of model built on subsample N_2, $N = N_1 + N_2$.

3. Determine F (this number is called a freedom of choice) best models using one of these criteria. The selected models y_i are then transferred to the second row of model construction. We search coefficients of new partial descriptions:

$$z_l = \phi^{(2)}(x_i, x_j) = a_0^{(2)} + a_1^{(2)} y_i + a_2^{(2)} y_j + a_3^{(2)} y_i^2 + a_4^{(2)} y_i y_j + a_5^{(2)} y_j^2$$

The process at the second row runs in the same way . The selection of the best models is carried out similarly, but $F_2 < F_1$. The process of rows construction repeats more and more till MSE (regularity criterion) falls. If at the m-th layer occurs the increase of the error $\bar{\varepsilon}^2$ the algorithm stops. In this case find the best model at the preceding layer and then moving backward by its connections find models of preceding layer and successfully passing all the used connections at the end we'll reach the first layer and find the analytical form of the optimal model (with minimal complexity).

2. Fuzzy GMDH. Principal ideas. Interval model of regression

Classical GMDH has some drawbacks:

1. GMDH utilizes least squared method (LSM) for finding the model coefficients but matrix of linear equations may be close to degenerate and the corresponding solution may appear non-stable and very volatile. Therefore, the special methods for regularization should be used;

2. after application of GMDH point-wise estimations are obtained but in many cases it's needed find interval value for coefficient estimates;

3. GMDH doesn't work in case of incomplete or fuzzy input data.

Therefore, in last 10 years the new variant of GMDH – fuzzy GMDH was developed and refined which may work with fuzzy input data and is free of classical GMDH drawbacks [Zaychenko, 2003; Zaychenko, 2006].

In works [Zaychenko, 2003; Zaychenko, 2006] the linear interval model regression was considered :

$$Y = A_0 Z_0 + A_1 Z_1 + \ldots + A_n Z_n \tag{6}$$

where A_i is a fuzzy number of triangular form described by pair of parameters $A_i = (\alpha_i, c_i)$, where α_i is interval center, c_i is its width, $c_i \geq 0$

Then *Y* is a fuzzy number, parameters of which are determined as follows:

the interval center

$$\alpha_y = \sum \alpha_i z_i = \alpha^T \cdot z, \tag{7}$$

the interval width

$$c_y = \sum c_i \cdot |z_i| = c^T |z|. \tag{8}$$

In order the interval is correct it's necessary that real value of output should belong to the interval of uncertainty described by the following constraints:

$$\begin{cases} \alpha^T z - c^T \cdot |z| \leq y \\ \alpha^T z + c^T \cdot |z| \geq y \end{cases}. \tag{9}$$

For example, for the partial description of the kind

$$f(x_i, x_j) = A_0 + A_1 x_i + A_2 x_j + A_3 x_i x_j + A_4 x_i^2 + A_5 x_j^2 \tag{10}$$

it's necessary to assign in the general model (6)

$$z_0 = 1, \; z_1 = x_i \; z_2 = x_j \; z_3 = x_i x_j \; z_4 = x_i^2 \; z_5 = x_j^2$$

Let the training sample be $\{z_1, z_2, ..., z_M\}$, $\{y_1, y_2, ..., y_M\}$. Then for the model (10) to be adequate it's necessary to find such parameters (α_i, c_i) $i = \overline{1,n}$, which satisfy the following inequalities:

$$\begin{cases} \alpha^T z_k - c^T \cdot |z_k| \le y_k \\ \alpha^T z_k + c^T \cdot |z_k| \le y_k \end{cases}, \quad k = \overline{1,M} . \tag{11}$$

Let's formulate the basic requirements for the linear interval model of partial description of a kind (10). It's necessary to find such values of the parameters (α_i, c_i) of fuzzy coefficients for which:

1. Real values of the observed outputs y_k would drop in the estimated interval for Y_k;
2. The total width of the estimated interval for all sample points would be minimal.

These requirements lead to the following linear programming problem:

$$\min(C_0 \cdot M + C_1 \sum_{k=1}^{M} |x_{ki}| + C_2 \sum_{k=1}^{M} |x_{kj}| + C_3 \sum_{k=1}^{M} |x_{ki} x_{kj}| + C_4 \sum_{k=1}^{M} |x_{ki}^2| + C_5 \sum_{k=1}^{M} |x_{kj}^2|, \tag{12}$$

under constraints:

$$\begin{aligned} &a_0 + a_1 x_{ki} + a_2 x_{kj} + a_3 x_{ki} x_{kj} + a_4 x_{ki}^2 + a_5 x_{kj}^2 - (C_0 + C_1 |x_{ki}| + C_2 |x_{kj}| + \\ &+ C_3 |x_{ki} x_{kj}| + C_4 |x_{ki}^2| + C_5 |x_{kj}^2|) \le \mathrm{y_k} \end{aligned} \tag{13}$$

$$\begin{aligned} &a_0 + a_1 x_{ki} + a_2 x_{kj} + a_3 x_{ki} x_{kj} + a_4 x_{ki}^2 + a_5 x_{kj}^2 + (C_0 + C_1 |x_{ki}| + C_2 |x_{kj}| + \\ &+ C_3 |x_{ki} x_{kj}| + C_4 |x_{ki}^2| + C_5 |x_{kj}^2|) \ge \mathrm{y_k} \\ &k = \overline{1,M} , \\ &C_p \ge 0, \quad p = 0,5 , \end{aligned} \tag{14}$$

where *k* is an index of a point.

As we can easily see the task (12) – (14) is linear programing (LP) problem. However, the inconvenience of the model (12) – (14) for the application of standard LP methods is that there are no constraints of non- negativity for variables α_i. Therefore for its solution it's reasonable to pass to the dual LP problem by introducing dual variables $\{\delta_k\}$ and $\{\delta_{k+M}\}$, $k = \overline{1,M}$. Using simplex- method for the dual problem and after finding the optimal values for the dual variables $\{\delta_k\}$ the optimal solutions (α_i, c_i) of the initial direct problem will be also found [Zaychenko, 2003; Zaychenko, 2006].

3. FGMDH with fuzzy input data for triangular membership functions

The generalization and further development of the considered FMGH is Fuzzy GMDH where fuzzy are not only model coefficients but input data as well. Below the correspondent mathematical model is presented. [Zaychenko, 2008]

3.1. The form of math model for triangular MF

Let's consider the linear interval regression model with fuzzy inputs which generalies the model (6) :

$$Y = A_0 Z_0 + A_1 Z_1 + \ldots + A_n Z_n, \tag{15}$$

where A_i – fuzzy number of triangular shape, which is described by threes of parameters $A_i = (\underline{A_i}, a_i, \overline{A_i})$, where a_i – center of the interval, $\overline{A_i}$ – its upper border, $\underline{A_i}$ - its lower border.

Current task contains the case of symmetrical membership function for parameters A_i, so they can be described via pair of parameters (a_i, c_i).

$$\underline{A_i} = a_i - c_i,\ \overline{A_i} = a_i + c_i,\ c_i \text{ – interval width, } c_i \geq 0,$$

Z_i – also fuzzy numbers of triangular shape, which are defined by parameters $(\underline{Z_i}, \breve{Z}_i, \overline{Z_i})$, $\underline{Z_i}$ - lower border, $\breve{Z}_i$ - center, $\overline{Z_i}$ - upper border of fuzzy number.

Then Y – fuzzy number, which parameters are defined as follows:

Center of the interval:

$$\breve{y} = \sum a_i * \breve{Z}_i,$$

Deviation in the left part of the membership function:

$$\breve{y} - \underline{y} = \sum (a_i * (\breve{Z}_i - \underline{Z_i}) + c_i \left|\breve{Z}_i\right|), \text{ thus}$$

Lower border of the interval:

$$\underline{y} = \sum(a_i * \underline{Z}_i - c_i \left|\breve{Z}_i\right|) \tag{16}$$

Deviation in the right part of the membership function:

$$\overline{y} - \breve{y} = \sum(a_i * (\overline{Z}_i - \breve{Z}_i) + c_i \left|\breve{Z}_i\right|) = \sum a_i \overline{Z}_i - a_i \breve{Z}_i + c_i \left|\breve{Z}_i\right| \text{, so}$$

Upper border of the interval:

$$\overline{y} = \sum(a_i * \overline{Z}_i + c_i \left|\breve{Z}_i\right|) \tag{17}$$

For the interval model to be correct, the real value of input variable Y should lay in the interval got by the method workflow.

It can be described in such a way:

$$\begin{cases} \sum(a_i * \underline{Z}_{ik} - c_i \left|\breve{Z}_{ik}\right|) \le y_k \\ \sum(a_i * \overline{Z}_{ki} + c_i \left|\breve{Z}_{ik}\right|) \ge y_k, k = \overline{1,M} \end{cases} \tag{18}$$

Where Z_k=[Z_k]i is input training sample, y_k –known output values, $k = \overline{1,M}$, M – number of observation points.

So, the general requirements to estimation linear interval model are to find such values of parameters (a_i, c_i) of fuzzy coefficients, which enable:

a) Observed values y_k lay in estimation interval for Y_k;

b) Total width of estimation interval is minimal.

These requirements can be redefined as a task of linear programming:

$$\min_{a_i, c_i} \sum_{k=1}^{M} (\sum(a_i * \overline{Z}_i + c_i \left|\breve{Z}_i\right|) - \sum(a_i * \underline{Z}_i - c_i \left|\breve{Z}_i\right|)) \tag{19}$$

under constraints:

$$\begin{cases} \sum(a_i * \underline{Z}_{ik} - c_i \left|\breve{Z}_{ik}\right|) \le y_k \\ \sum(a_i * \overline{Z}_{ki} + c_i \left|\breve{Z}_{ik}\right|) \ge y_k, k = \overline{1,M} \end{cases} \tag{20}$$

3.2. Formalized problem formulation in case of triangular membership functions

Let's consider partial description

$$f(x_i, x_j) = A_0 + A_1 x_i + A_2 x_j + A_3 x_i x_j + A_4 x_i^2 + A_5 x_j^2 \tag{21}$$

Then math model (19)-(20) takes the form

$$\begin{aligned}\min_{a_i, c_i} \; & (2Mc_0 + a_1 \sum_{k=1}^{M} (\bar{x}_{ik} - \underline{x}_{ik}) + 2c_1 \sum_{k=1}^{M} |\breve{x}_{ik}| + a_2 \sum_{k=1}^{M} (\bar{x}_{jk} - \underline{x}_{jk}) + 2c_2 \sum_{k=1}^{M} |\breve{x}_{jk}| + \\ & + a_3 \sum_{k=1}^{M} (|\breve{x}_{ik}| (\bar{x}_{jk} - \underline{x}_{jk}) + |\breve{x}_{jk}| (\bar{x}_{ik} - \underline{x}_{ik})) + 2c_3 \sum_{k=1}^{M} |\breve{x}_{ik} \breve{x}_{jk}| + 2a_4 \sum_{k=1}^{M} |\breve{x}_{ik}| (\bar{x}_{ik} - \underline{x}_{ik}) + \\ & + 2c_4 \sum_{k=1}^{M} \breve{x}_{ik}^2 + 2a_5 \sum_{k=1}^{M} |\breve{x}_{jk}| (\bar{x}_{jk} - \underline{x}_{jk}) + 2c_5 \sum_{k=1}^{M} \breve{x}_{jk}^2)\end{aligned} \tag{22}$$

with the following conditions:

$$\begin{aligned} & a_0 + a_1 \underline{x}_{ik} + a_2 \underline{x}_{jk} + a_3 (-|\breve{x}_{ik}| (\breve{x}_{jk} - \underline{x}_{jk}) - |\breve{x}_{jk}| (\breve{x}_{ik} - \underline{x}_{ik}) + \breve{x}_{ik} \breve{x}_{jk}) + \\ & + a_4 (-2 |\breve{x}_{ik}| (\breve{x}_{ik} - \underline{x}_{ik}) + \breve{x}_{ik}{}^2) + a_5 (2 |\breve{x}_{jk}| (\breve{x}_{jk} - \underline{x}_{jk}) + \breve{x}_{jk}{}^2) - c_0 - c_1 |\breve{x}_{ik}| - \\ & - c_2 |\breve{x}_{jk}| - c_3 |\breve{x}_{ik} \breve{x}_{jk}| - c_4 \breve{x}_{ik}^2 - c_5 \breve{x}_{jk}^2 \le y_k \\ & a_0 + a_1 \bar{x}_{ik} + a_2 \bar{x}_{jk} + a_3 (|\breve{x}_{ik}| (\bar{x}_{jk} - \breve{x}_{jk}) + |\breve{x}_{jk}| (\bar{x}_{ik} - \breve{x}_{ik}) - \breve{x}_{ik} \breve{x}_{jk}) + a_4 (2 |\breve{x}_{ik}| (\bar{x}_{ik} - \\ & - \breve{x}_{ik}) - \breve{x}_{ik}{}^2) + a_5 (2 |\breve{x}_{jk}| (\bar{x}_{jk} - \breve{x}_{jk}) - \breve{x}_{jk}{}^2) + c_0 + c_1 |\breve{x}_{ik}| + c_2 |\breve{x}_{jk}| + c_3 |\breve{x}_{ik} \breve{x}_{jk}| + \\ & c_4 \breve{x}_{ik}^2 + c_5 \breve{x}_{jk}^2 \ge y_k \end{aligned} \tag{23}$$

$$c_l \ge 0, \; l = \overline{0,5}.$$

As we can see, this is the linear programming problem, like the problem (12)-(13)for non-fuzzy inputs but there are still no limitations for non-negativity of variables a_i, so we need go to dual problem, introducing dual variables $\{\delta_k\}$ and $\{\delta_{k+M}\}$.

Write down dual problem:

$$\max\left(\sum_{k=1}^{M} y_k \cdot \delta_{k+M} - \sum_{k=1}^{M} y_k \cdot \delta_k\right) \tag{24}$$

Under constraints:

$$\begin{aligned} & \sum_{k=1}^{M} \delta_{k+M} - \sum_{k=1}^{M} \delta_k = 0 \\ & \sum_{k=1}^{M} \bar{x}_{ik} \cdot \delta_{k+M} - \sum_{k=1}^{M} \underline{x}_{ik} \cdot \delta_k = \sum_{k=1}^{M} (\bar{x}_{ik} - \underline{x}_{ik}) \end{aligned} \tag{25}$$

$$\sum_{k=1}^{M} \bar{x}_{jk} \cdot \delta_{k+M} - \sum_{k=1}^{M} \underline{x}_{jk} \cdot \delta_k = \sum_{k=1}^{M} (\bar{x}_{jk} - \underline{x}_{jk})$$

$$\sum_{k=1}^{M} (|\breve{x}_{ik}|(\bar{x}_{jk} - \breve{x}_{jk}) + |\breve{x}_{jk}|(\bar{x}_{ik} - \breve{x}_{ik}) - \breve{x}_{ik}\breve{x}_{jk}) \cdot \delta_{k+M} -$$
$$-\sum_{k=1}^{M} (-|\breve{x}_{ik}|(\breve{x}_{jk} - \underline{x}_{jk}) - |\breve{x}_{jk}|(\breve{x}_{ik} - \underline{x}_{ik}) + \breve{x}_{ik}\breve{x}_{jk}) \cdot \delta_k =$$
$$= \sum_{k=1}^{M} (|\breve{x}_{ik}|(\bar{x}_{jk} - \underline{x}_{jk}) + |\breve{x}_{jk}|(\bar{x}_{ik} - \underline{x}_{ik}))$$

$$\sum_{k=1}^{M} (2|\breve{x}_{ik}|(\bar{x}_{ik} - \breve{x}_{ik}) - \breve{x}_{ik}^{\,2}) \cdot \delta_{k+M} - \sum_{k=1}^{M} (-2|\breve{x}_{ik}|(\breve{x}_{ik} - \underline{x}_{ik}) + \breve{x}_{ik}^{\,2}) \cdot \delta_k = \sum_{k=1}^{M} |\breve{x}_{ik}|(\bar{x}_{ik} - \underline{x}_{ik})$$

$$\sum_{k=1}^{M} (2|\breve{x}_{jk}|(\bar{x}_{jk} - \breve{x}_{jk}) - \breve{x}_{jk}^{\,2}) \cdot \delta_{k+M} - \sum_{k=1}^{M} (-2|\breve{x}_{ik}|(\breve{x}_{jk} - \underline{x}_{jk}) + \breve{x}_{jk}^{\,2}) \cdot \delta_k = \sum_{k=1}^{M} |\breve{x}_{jk}|(\bar{x}_{jk} - \underline{x}_{jk})$$

$$\sum_{k=1}^{M} \delta_{k+M} + \sum_{k=1}^{M} \delta_k \le 2M$$

$$\sum_{k=1}^{M} |\breve{x}_{ik}| \cdot \delta_{k+M} + \sum_{k=1}^{M} |\breve{x}_{ik}| \cdot \delta_k \le 2\sum_{k=1}^{M} |\breve{x}_{ik}|$$

$$\sum_{k=1}^{M} |\breve{x}_{jk}| \cdot \delta_{k+M} + \sum_{k=1}^{M} |\breve{x}_{jk}| \cdot \delta_k \le 2\sum_{k=1}^{M} |\breve{x}_{jk}|$$

$$\sum_{k=1}^{M} |\breve{x}_{ik}\breve{x}_{jk}| \cdot \delta_{k+M} + \sum_{k=1}^{M} |\breve{x}_{ik}\breve{x}_{jk}| \cdot \delta_k \le 2\sum_{k=1}^{M} |\breve{x}_{ik}\breve{x}_{jk}|$$

$$\sum_{k=1}^{M} \breve{x}_{ik}^2 \cdot \delta_{k+M} + \sum_{k=1}^{M} \breve{x}_{ik}^2 \cdot \delta_k \le 2\sum_{k=1}^{M} \breve{x}_{ik}^2 \qquad (26)$$

$$\sum_{k=1}^{M} \breve{x}_{jk}^2 \cdot \delta_{k+M} + \sum_{k=1}^{M} \breve{x}_{jk}^2 \cdot \delta_k \le 2\sum_{k=1}^{M} \breve{x}_{jk}^2$$

$$\delta_k \ge 0 \ , \delta_{k+M} \ge 0 \ , \ k = \overline{1,M} \qquad (27)$$

The task (24)-(27) can be solved using simplex-method. Having optimal values of dual variables $\{\delta_k\}$, $\{\delta_{k+M}\}$, we easily obtain the optimal values of desired variables c_i, a_i, $i = \overline{0,5}$, and also a desired fuzzy model for given partial description.

4. The description of fuzzy algorithm GMDH

Let's present the brief description of the algorithm FGMDH [Zaychenko, 2006].

1. Choose the general model type by which the sought dependence will be described.
2. Choose the external criterion of optimality (criterion of regularity or non --biasedness).
3. Choose the type of partial descriptions (for example, linear or quadratic one).
4. Divide the sample into training N_{train} and test N_{test} subsamples.
5. Put zero values to the counter of model number k and to the counter of rows r (iterations number).
6. Generate a new partial model f_k (10) using the training sample. Solve the LP problem (12) – (14) or (22)-(23) and find the values of parameters α_i , c_i .
7. Calculate using test sample the value of external criterion ($N_{ubk}^{(r)}$ or $\delta_k^{(2)}(r)$).
8. $k = k + 1$. If $k > C_N^2$ for r=1or $k > C_F^2$ for r>1, then $k = 1$, $r = r + 1$ and go to step 9, otherwise go to step 6.
9. Calculate the best value of the criterion for models of r-th iteration. If $r = 1$, then select F best models and assigning $r = r + 1$, $k = 1$, go to step 6 and execute (r+1)-th iteration otherwise, go to step 10.
10. If $\left|N_{ub}(r) - N_{ub}(r-1)\right| \le \varepsilon$ or $\delta_k^{(2)}(r) \ge \delta_{k-1}^{(2)}(r)$, then go 11,
11. Otherwise select F best models and assigning $r = r + 1$, $k = 1$, go to step 6 and execute (r+1) iteration.
12. Select the best model out of models of the previous row (iteration) using external criterion.

Starting from this model and moving backward by its connection to the models of previous row and successively passing the models of all previous rows by corresponding connections at the last step reach the models of the first row. Having made corresponding reverse substitutions of variables we find the final best model in initial variables $Y = F(x_1, x_2, ..., x_n)$.

Thus, fuzzy GMDH allows to construct fuzzy models and has the following advantages:

1. The problem of ill- conditionality of matrix of normal equalities is absent in fuzzy GMDH unlike classic GMDH as the least squared method isn't used for optimal model determination. The problem of optimal model determination is transferred to the problem of linear programming, which is always solvable.
2. There is interval regression model built as the result of method work unlike GMDH which constructs point-wise models. And the interval width enables to estimate the accuracy of the found model.
3. There is a possibility of the obtained model adaptation.

5. The application of GMDH for forecasting at the stock exchange

For estimation of efficiency of the suggested FGMDH method with non-fuzzy and fuzzy inputs the corresponding software kit was elaborated and numerous experiments of financial markets forecasting were carried out. For the experiments the stock prices of different shares at the Stock exchange "RTS" were chosen. Some of them are presented below

Experiment 1. RTS-2 index forecasting (opening price)

There were 5 fuzzy input variables in this experiment; they were price on shares of "second echelon" Russian energetic companies, which are included to RTS-2 index computation list:

BANE – shares of "Башнефть" joint-stock company,

ENCO – shares of "Сибирьтелеком" joint-stock company,

ESMO – shares of "ЦентрТелеком" joint-stock company,

IRGZ – shares of "Иркутскэнерго" joint-stock company,

KUBN – shares of "Южтелеком" joint-stock company.

Output variable is the value of RTS-2 index (opening price) for the same period (03.04.2006 – 18.05.2006).

Sample size – 32 values.

Training sample size – 19 values (optimal size of training sample for current experiment).

The following results were obtained:

1. **For triangular membership function**

   Criterion for this experiment was MSE=0,061787

   The corresponding results are presented at the Figure 3.

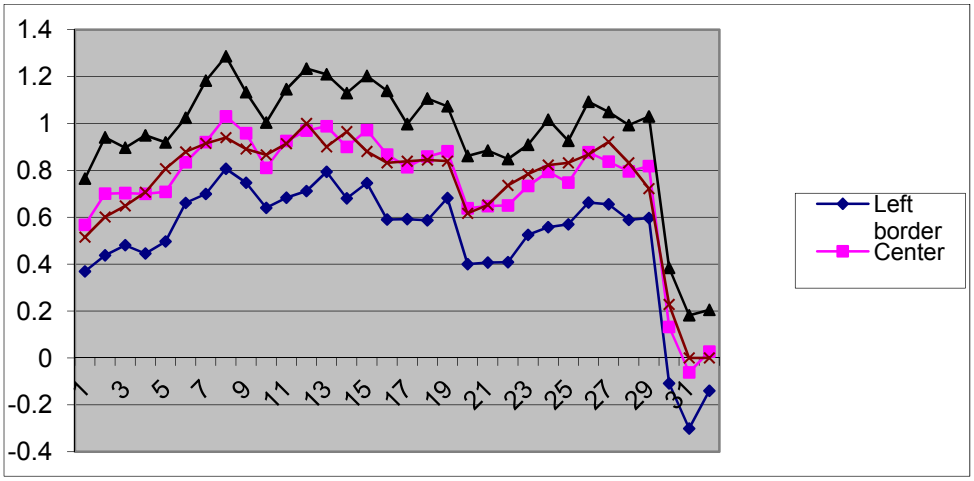


Figure 3. Experiment 1 result for triangular MF and normalized values of input variables

a) For non-normalized input data

Criterion value for this experiment was:

MSE = 6,407928

MAPE =0,24%

2. For Gaussian membership function (optimal level α=0,85)

a) For normalized input data

Criterion value: MSE = 0,033097.

The corresponding results are presented at the Figure 4.

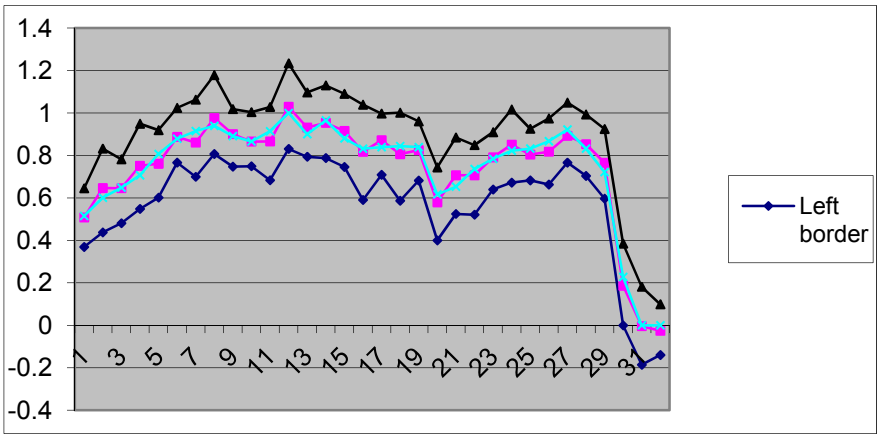


Figure 4. Experiment 1 result for Gaussian MF and normalized values of input variables

b) For non-normalized input data

Criterion value: MSE = 3,432511 MAPE =0,13%

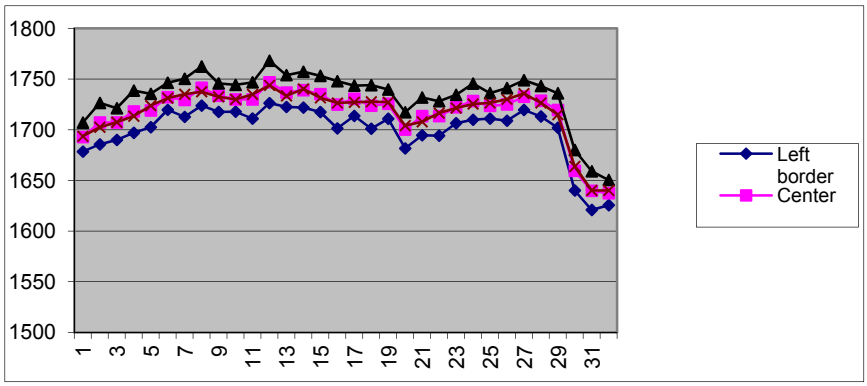


Figure 5. Experiment 1 result for Gaussian MF and non-normalized values of input variables

The total results for triangular and Gaussian MF are presented in the table 1.As we can see from the presented results of experiment 1, forecasting using triangular and Gaussian membership functions gives good results. Results of experiments with Gaussian MF are better than results of experiments with triangular MF.

Table 1. Forecasting results at RTS stock exchange

For non-normalized data	Triangular MF	Gaussian MF
MSE	0,061787	0,033097
For normalized data	Triangular MF	Gaussian MF
MSE	6,407928	3,432511
MAPE	0,24%	0,13%

6. The comparison of GMDH, FGMDH and FGMDH with fuzzy inputs

In the next experiments the comparison of the suggested method FGMDH with fuzzy inputs with known methods: classical GMDH and Fuzzy GMDH was performed

Experiment 2. Forecasting of RTS index (opening price)

Current experiment contains 5 fuzzy input variables, which are the stock prices of leading Russian energetic companies included into the list of RTS index calculation:

Output variable is the value of RTS index (opening price) of the same period (03.04.2006 – 18.05.2006).

Sample size – 32 values.

Training sample size – 18 values (optimal size of the training sample for current experiment).

The following results were obtained presented at the Table 2 and Fig. 6.

Table 2. MSE comparison for different methods of experiment 2

	GMDH	FGMDH	FGMDH with fuzzy inputs, Triangular MF	FGMDH with fuzzy inputs, Gaussian MF
MSE	0,1129737	0,0536556	0,055557	0,028013

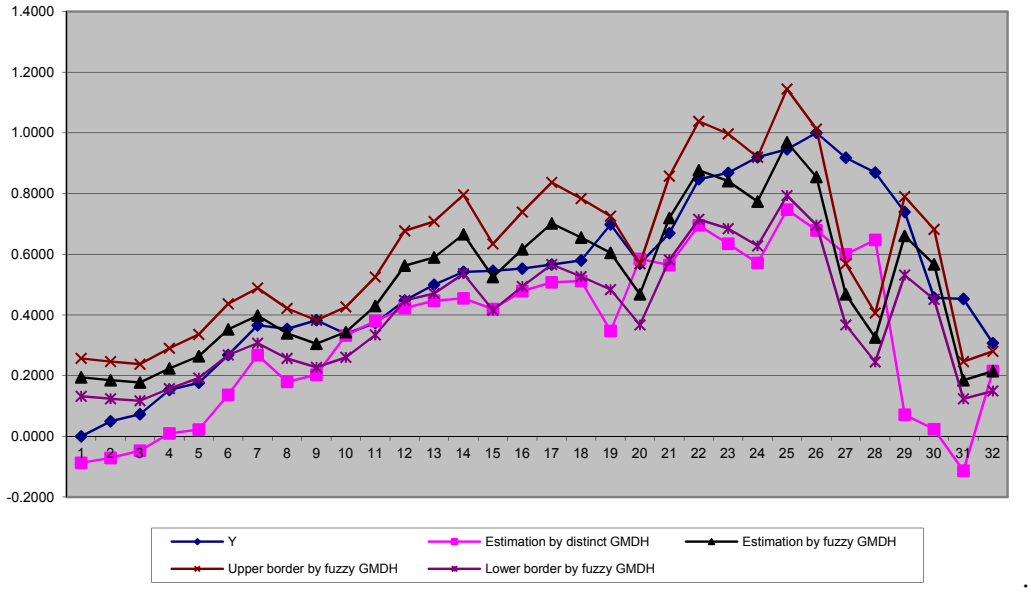


Figure 6. Experiment 2 results using GMDH and FGMDH

As the results of experiment 2 show, fuzzy group method of data handling with fuzzy input data gives more accurate result than FGMDH with triangular membership function or Gaussian membership function. In case of triangular MF FGMDH with fuzzy data gives a little worse than FGMDH with Gaussian MF.

Experiment 3. RTS-2 index forecasting (closing price)

Sample size – 32 values.

Training sample size – 19 values (optimal size of training sample for current experiment).

The following results were obtained, which are presented in Table 3.

Table 3. MSE of different methods of experiment 3 comparison

	GMDH	FGMDH	FGMDH with fuzzy inputs, triangular MF	FGMDH with fuzzy inputs, Gaussian MF
MSE	0,051121	0,063035	0,061787	0,033097

As the results of the experiment 4 show, fuzzy group method of data handling with fuzzy input data gives the better result than GMDH and FGMDH in case of Gaussian membership functions. At the same

time in this experiment GMDH gives the better results, than FGMDH and FGMDH with fuzzy input data in the case of triangular membership functions.

Experiment 4. RTS index forecasting (opening price)

For the efficiency estimation stock indexes forecasting using fuzzy neural nets (FNN) with Mamdani and Tsukamoto algorithms were carried out. Total 267 everyday indexes of stock prices during period from 1.04.2005 to 30.12.2005 were used for neural net training. The following results were obtained

Table 4. Experiment 4 results using FNN

Criterion	Mamdani with Gaussian MF	Mamdani with Triangular MF	Tsukamoto with Gaussian MF	Tsukamoto with Triangular MF
MSE	3,692981	3,341179	7,002467	5,119318
MAPE %	0,256091	0,318056	0,318056	0,419659

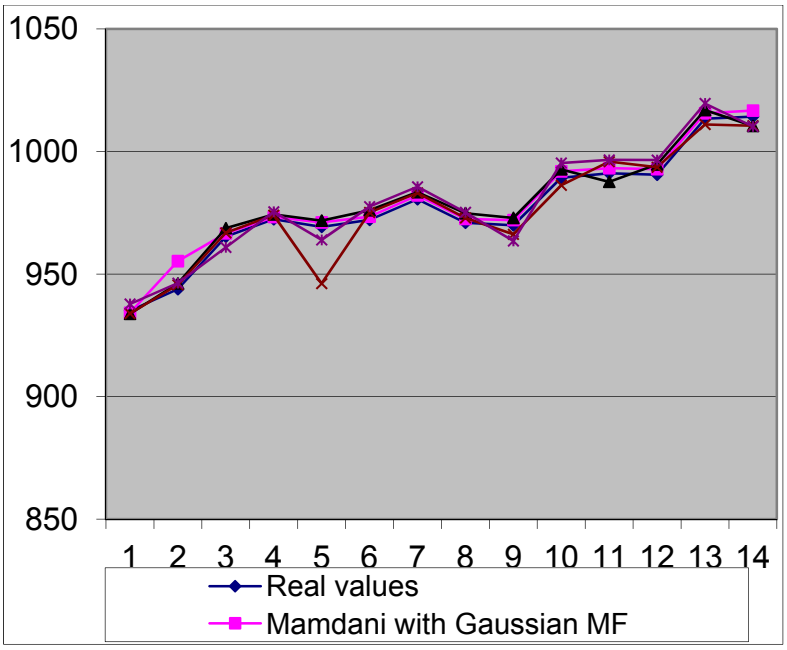


Figure 7. Experiment 4 forecasting results using FNN

As experiment 4 results show, forecasting using Mamdani controller with Gaussian MF was the best, Mamdani controller with triangular MF is on the second place.

The comparative results of forecasting accuracy of FNN and different variants of FGMDH were carried out. The corresponding results are presented at the Table 5 and Figure 8.

Table 5. Forecasting results for FGMDH and FNN

	Mamdani controller	Tsukamoto Controller	FGMDH With fuzzy Inputs (4 input variables)	FGMDH with fuzzy inputs (previous values of forecasted variable used)
MSE for Gaussian MF	0,18046	0,26801	0,115072	0,094002
MSE for triangular MF	0,28112	0,34443	0,210865	0,215421

The best MSE was achieved by FGMDH with fuzzy inputs, and this method also allows to build interval estimation of the forecasted value. FGMDH with fuzzy inputs using Gaussian MF gives more accurate forecast than triangular MF as well as with FNN.

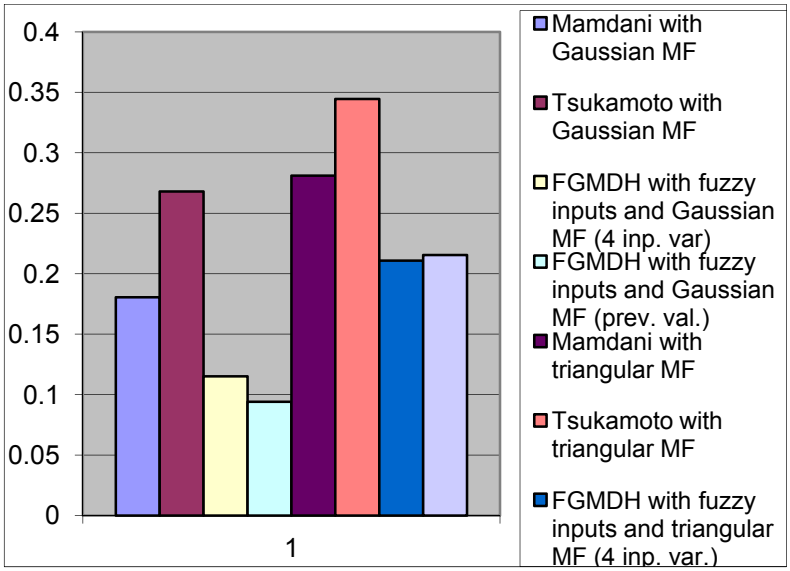


Figure 8. MSE comparison for FMGH and FNN

Experiment 5. "Lukoil" stock prices forecasting based on previous data about stock prices of leading Russian energetic companies for the same period.

Input variables:

EESR – shares of "РАО ЕЭС России" joint-stock company,

YUKO – shares of "ЮКОС" joint-stock company,

SNGSP – privileged shares of "Сургутнефтегаз" joint-stock company,

SNGS – common shares of "Сургутнефтегаз" joint-stock company.

The results are presented at the Table 6.

Table 6. Forecasting results in experiment 4.

		Mamdani Controller	Tsukamoto Controller	FGMDH with fuzzy inputs (4 input variables)	FGMDH with fuzzy inputs (previous values on the input)
Gaussian MF	MSE	3,692981	7,002467	2,1151183	2,886697
	MAPE, %	0,256091	0,318056	0,179447	0,256547
	MSE	3,34179	5,119318	4,717268	4,977901
triangular MF	MAPE, %	0,318056	0,419659	0,40437	0,415434

As current experiment results show, forecasting using FGMDH with fuzzy input data using Gaussian membership function was the best, fuzzy Mamdani controller with Gaussian MF is on the second place.

In a whole the experiments have shown the high accuracy of forecasting using FGMDH in comparison with FNN. The additional advantage od GMDH is its possibility to work with short samples and under uncertainty when input data are fuzzy.

Conclusion

1. The problem of finding unknown dependencies in big data was considered. For its solution inductive modeling method GMDH was suggested which allows constructing models with unknown structure almost automatically. Besides GMDH may work with insufficient data available (Short samples).
2. In case of incomplete or unreliable data fuzzy GMDH with fuzzy inputs was suggested for synthesis of corresponding forecasting models in experimental data.
3. The experimental investigations of the suggested method in the problem of stock prices forecasting with different types of partial descriptions were carried out.
4. The comparison of forecasting accuracy of FGMDH and fuzzy neural networks Mamdani and Tsukamoto was performed confirming the efficiency of FGMDH.

Bibliography

[Barsegyan, 2008] BarsegyanA.A. Technologies of data analysis: Data mining, Visual Mining, TextMining, OLAP. / A.A. Barsegyan, M.C. Kuprianov, V.V. Stepanenko, I.I. Holod.-. -2-nd edition, revised and add.).- SPb.: BHV- Petersburg, 2008.- 384 p. (rus)

[Bodyanskiy, 2009] Bodyanskiy Ye, Zaychenko Yu., Pavlikovskaya E., Samarina M., Viktorov Ye. Neo-fuzzy neural network structure optimization using GMDH for solving forecasting and classification problems // Proc. Int. Workshop on Inductive Modeling 2009. Krynica,Poland, 2009.- 77- 89.

[Duke, 2001] V. Duke, A. Samoilenko. Data Mining: Learning course. Publ. House " Peter". Moscow, Saint- Petersburg, Kharkov, Minsk, 2001.- 366 p. (rus)

[Ivakhnenko, 1985] Ivakhnenko A.G., Mueller I.A. Self-organization of forecasting models.-Kiev: Publ. House "Technika".– 1985. (rus)

[Zaychenko, 2003]Zaychenko Yu. P. Fuzzy Group Method of Data Handling // System research and information technologies.-2003.-№3.-pp.-25-45. (rus)

[Zaychenko, 2006] Zaychenko Yu. The Fuzzy Group Method of Data Handling and Its Application for Economical Processes forecasting // Scientific Inquiry , vol.7, No 1, June, 2006.-pp. 83-98.

[Zaychenko, 2008] Zaychenko Yu. The Fuzzy Group Method of Data Handling with Fuzzy Input Variables // Scientific Inquiry , vol.9, No 1, June, 2008.-pp. 61-76.

Acknowledgement

The paper is published with financial support by the project ITHEA XXI of the Institute of Information Theories and Applications FOI ITHEA (www.ithea.org) and the Association of Developers and Users of Intelligent Systems ADUIS Ukraine (www.aduis.com.ua).

Authors' Information

Yuri Zaychenko *– Professor, doctor of technical sciences, Institute for applied system analysis, NTUU "KPI", 03056, Ukraine, Kyiv, Peremogi pr. 37, Corpus 35; e-mail: baskervil@voliacable.com, zaychenkoyuri@ukr.net*

Major Fields of Scientific Research: Information systems, Fuzzy logic, Decision making theory

Galib Hamidov- *PhD, Azarishig, Head of the Information technologies department, Baku, Azerbaijan , Galib.hamidov@bes.az*

Major Fields of Scientific Research: Information technologies, Data Mining

CROSS-PLATFORM ENVIRONMENT FOR APPLICATION LIFE CYCLE MANAGEMENT

Elena Chebanyuk, Oleksii Hlukhov

Abstract: *"Application Lifecycle Management (ALM) integrates and governs the planning, definition, design, development, testing, deployment, and management phases throughout the application lifecycle" [OMG, 2006].*

This paper is devoted to designing of ALM for supporting all software development processes. A review of papers, making strong contribution for improving software development life cycle processes is represented. This review touches three branches of investigation, namely papers, related to: (1) improving of communication processes between stakeholders; (2) increasing effectiveness of some operations in software development life cycle processes; (3) developing fundamental methods and tools for performing different operations related to several software development life cycle. Then comparative analysis of such ALM environments as Visual Studio, Team Foundation Server, FusionForge, TeamForge, IBM Rational Team Concert, IBM Rational Software Architect, and IBM Rational Functional Tester, is performed. Comparison of different ALM environments' functionality lets to formulate requirements for designing cross-platform ALM environment.

Then the conceptual schema of cross-platform ALM based on Eclipse environment is proposed. All plugins' functionalities were properly tested. Collaboration of plugins for supporting several software development tasks is accurately defined. Data flows for different plug-ins collaboration are shown. These data flows are considered for several kinds of stakeholders roles.

In conclusion, recommendations for raising effectiveness of software development life cycle processes, using proposed cross-platform ALM environment, are presented.

Keywords: *software development life cycle, application life cycle management, software development life cycle process, requirement analysis, software designing, software testing, deployment, Eclipse, Team Foundation Server, FusionForge, TeamForge, IBM Rational Team Concert, IBM Rational Software Architect, IBM Rational Functional Tester, Mylyn, Javadoc, JUnit, STAN, FindBugs, Jubula, UML to Java Generator, Eclipce IDE, Data tools platform, windows builder, Eclipce color theme, RMF, UML Designer.*

ITHEA Classification Keywords: *D.2.1 Requirements/Specifications (D.3.1) - Elicitation methods (e.g., rapid prototyping, interviews, JAD, Methodologies (e.g., object-oriented, structured), Tools; D.2.5 Testing and Debugging - Testing tools (e.g., data generators, coverage testing); D.2.6 Programming Environments - Integrated environments; D.2.9 Management - Life cycle; D.2.11 Software Architectures; D.2.13 Reusable Software*

Introduction

According to OMG definition: "An Application Lifecycle is the continuum of activities required to support an enterprise application from its initial inception through its deployment and system optimization"[OMG, 2006].

"Application Lifecycle Management (ALM) integrates and governs the planning, definition, design, development, testing, deployment, and management phases throughout the application lifecycle" [OMG, 2006].

In software development process today incremental-iterative software development approaches is implemented.

New challenges for improvement effectiveness of software development life cycle processes causes to modifying existing techniques and tools for increasing of their effectiveness. In order to reach this goal application performance management (APM) tools are involved in software development life cycle process.

Application performance management (APM) tools offer these capabilities, enabling companies to diagnose problems quickly and improve service quality. For companies that are using Agile and DevOps processes, APM can help improve communication and expedite software delivery. It enables continuous monitoring and testing during all phases of software delivery, including production [IBM, 2015].

Certifying organizations like the International Standards Organization (ISO) have effectively worked on various models and suggested guidelines, procedures that may be adopted by IT vendors. Most of these models have focused on process improvements [Misra, 2017].

Related papers

To improve existing APM tools it is necessary to investigate software development life cycle processes and activities of stakeholders' collaboration.

It is a precondition of appearing many scientific papers and vendor solutions addressed to solve this topic. Such papers are divided on two directions, namely to improving collaboration between stakeholders and features of software development life cycle processes.

Consider result of researches, directed to investigating processes of improving collaboration between stakeholders.

Paper [Misra, 2017] proposes to estimate user capabilities depending on their roles in software development process. User capabilities are identified in two categories: IT users who are IT experts and involved in design, development, and implementation of SDLC driven projects, and, second, non-IT users who, despite having inadequate or no exposure to IT, contribute to SDLC driven projects. The framework is implemented through Unified Modeling Language (UML) based approach. Paper contains detailed analysis of stakeholder's activities in every software development life cycle process. These roles are primarily end-users, planners, and domain experts (IT and non-IT). For every user role it is defined which UML diagrams solve tasks of such kind of users the best.

During the early stage of IT acquisition, managing IT activities relating to operation, programming, and data collection were the major areas of concern. In later stages the focus was on establishing a unit to look after various types of applications over an extended lifecycle, despite change in technology.

Authors also note that most organizations use different life cycle models for different projects. However, it is difficult to ascertain the survivability of the system thus developed for its expected life cycle. It is argued that most of the models popularly coming under SDLC have limitations in delivering good result in a complex scenario, but are successful in a tightly specified domain. All software models under SDLC can be characterized as a problem solving loop, which may go through four distinct stages: status quo, problem definition, solution integration, post-acquisition assessment.

Thus, they propose recommendation for improving software development life cycle processes. Also challenges for increasing of software development life cycle processes effectiveness are formulated.

Software development life cycle processes are the brick from which software development life cycle is consists. Successful management of all software development life cycle process is a very complex task due to the next causes:

- When software requirements changes other software development artifacts are changed too;
- Tools, platform and software environments are changing very quickly too. Thus, time to study and obtain practical skills with new environments is needed;
- Changing of technologies in turn leads to changing of some actions in performing software development processes;
- Some vendors adopt classical schemes [OMG, 2006].

Consider papers, relating to challenges of designing effective ALM and improving technologies or tools for performing different ALM tasks.

The paper [Grichi, 2015] deals with the verification of reconfigurable real-time systems to be validated by using the Object Constraint Language (abbrev, OCL). Authors propose an extension of OCL, named Reconfigurable OCL, in order to optimize the specification and validation of constraints related to different execution scenarios of a flexible system.

Also a metamodel of the new ROCL is proposed with formal syntax and semantics. This solution gains in term of the validation time and the quick expression of constraints.

But papers lack recommendation about ROCL implementing to increase effectiveness of software development lifecycle processes. Also software tools, supporting designed OCL extension, were not described.

Paper [Chebanyuk and Markov, 2016] presents an approach, verifying class diagram correspondence to SOLID object oriented design principles, is proposed in this paper. SOLID is an acronym, encapsulating the five class diagram design principles namely: Single responsibility, Open closed, Liskov substitution, Interface segregation and Dependency inversion.

To check whether class diagram meets to SOLID, its analytical representation is analyzed by means of predicate expressions. Analytical representation describes interaction of class diagram constituents, namely classes and interfaces, in set-theory terms. For every SOLID design principle corresponded predicate expressions are proposed. Also criteria for estimation of analysis results are formulated. But paper lacks representing this approach in some restriction language.

Paper [Chebanyuk, 2014] presents a method of behavioral software models synchronization. Implementing this method behavioral software models, which are changed after communication with customer, are synchronized with other software models that are represented as UML diagrams. Method of behavioral software artifacts synchronization makes the Model-Driven Development (MDD) approach more effective. For synchronization of different behavioral software models, transformation approach in the area of Model-Driven Architecture (MDA) is proposed. Synchronization operation is executed using analytical representation of initial and resulting models. Initial behavioral software model is represented by UML Use Case Diagram. Resulting behavioral software model is represented as UML Collaboration Diagram. Analytical representation of UML Use Case diagram allows considering data flows. For this representation set-theory tool operations are used. As a Collaboration Diagram usually contains more information in comparison with Use Case one, method defines two types of Use Case diagram fragments. From the beginning Use Case diagram fragments that can be transformed directly to resulting diagram constituents are considered. Then the rest of Use Case diagram fragments are processed to represents rules of placement Collaboration Diagram messages. These rules help to designate data flows, represented in Collaboration Diagram, more accuracy. Method, proposed in this

article, can be used both separately and be a part of more complex transformation technics, methods and frameworks solving different tasks in MDA sphere.

Paper [Filho, 2016] presents an approach for resource identification, management, and service discovery in Service Oriented Architecture (SOA). The service identification process consists of a combination of top-down and bottom-up techniques of domain decomposition and existing asset analysis. In the top-down view, a blueprint of business use cases provides the specification for business services. In the bottom-up approach, the analyst departs from a service identifying the provider entity, where application and container it is located. The idea is to reach a context view from a service. In order to reuse a service, clients need to know much more than a simple service name or the address of the service provider. Developers need to see a service as an interface, including methods that they will invoke in order to execute the service and their necessary parameters. The lookup service can be seen as a directory service, where services are found and viewed.

The descriptors specifications include: *(i)* the service name (the entity type that provides the service); *(ii)* the path (URL) where the service is allocated; *(iii)* the scope informing if the service is local (in the container) or remote; *(iv)* name and type of the parameters; *(v)* a brief description of the service functionality; *(vi)* a return informing if any data type returns to the caller service; *(vii)* the keywords related to the service; and *(viii)* implementation, informing if the service is: implemented in Java, a Web service, or a legacy service encapsulated as a service in a component [Filho, 2016].

The approach emphasizes an architectural model that allows representation, description, and identification of services, and is explored as a metadata repository. It is focused not only on Web Services, but also in all services existing in big companies' applications, including currently developed services and legacy system services, highlighting the importance of reusing fine granularity services. The model includes discovery procedures to find and retrieve candidates for services composition and reuse. These procedures adopt a Case-Based Reasoning approach, in which the services are considered as cases kept and indexed in a repository. Case matching is carried out by means of text mining techniques that allow finding the most appropriate service candidate with the desired requirements for a particular task.

Authors propose to store services in local store and describe service in WSDL format. The next scheme of service preparation is proposed

The process starts with a description of a service required by a developer. This description includes a functional account of the service representing the developer experience, beyond the usual descriptors like *name* and *parameters*. The system searches for a service in CB, guided by a given description, and then it retrieves a list of the best matching services. If a service satisfies the developer necessity, than it is applied. Otherwise, the alternatives are: *(i)* to search in the Web, *(ii)* to adapt a case from the

retrieved case list, or *(iii)* to develop a new solution. In any case, the case base must be updated [Filho, 2016].

But the problem of remote user: obtaining information about service if its functionality or address is changed. One of the solutions is to deploy additional data storage for services identification.

As requirement analysis is a very important process consider papers, related improvement quality of operations, performed in it.

Paper [Shamra, 2014] proposes an approach of generation sequence or activity diagrams from requirements, presented in natural text. Requirements analysis process involves developing abstract models for the envisioned or the proposed software system. However, software requirements are captured in the form of Natural Language and, generating UML models from natural language requirements relies heavily on individual expertise. Thus, authors present an approach towards automated generation of behavioral UML models, namely activity diagrams and sequence diagrams. Initial information for transformation is lexical and syntactic analysis of NL statements that is grounded on patterns. Authors propose an idea to analyze requirement specification involving natural language patterns.

Patterns – grammatical knowledge or domain-specific prove helpful in improving the quality of analysis. Knowledge patterns are divided on three types: lexical patterns for indicating a relation; grammatical patterns, which are combinations of part-of-speech; and, paralinguistic patterns, which include punctuation, parenthesis, text structure etc. [Shamra, 2014].

The idea of approach is based on transforming the requirements statements to intermediary structured representations – frames. Frames are special structures for representing knowledge using slots to store knowledge in an Object Oriented manner and, are an efficient means for reasoning. Then frames are translated to behavioral UML models. Authors use peculiarities of constructing sentences in English language, namely parts of sentences, times and verb forms, as well as examples of frames for composing prepositions, active, and passive voice. For analysis of sentence Stanford Dependency parser is used [Stanford, 2015].

Knowledge also are stored in frames is then used to automatically generate activity and sequence diagram. Also authors use common in activity and sequence diagram elements notations.

Paper [Inoue, 2015] proposes an extension of goal graphs in goal-oriented requirements engineering. First it is necessary to understanding the relations between goals. Goals specify multiple concerns such as functions, strategies, and non-functions, and they are refined into sub goals from mixed views of these concerns. This intermixture of concerns in goals makes it difficult for a requirements analyst to understand and maintain goal graphs. In our approach, a goal graph is put in a multi-dimensional space,

a concern corresponds to a coordinate axis in this space, and goals are refined into sub goals referring to the coordinates. Thus, the meaning of a goal refinement is explicitly provided by means of the coordinates used for the refinement. By tracing and focusing on the coordinates of goals, requirements analysts can understand goal refinements and modify unsuitable ones.

Paper [Escande, 2013] proposed method of defining requirement priority and assigning it to different stakeholders. Requirements are proposed to be assigned to different levels of priorities. Then, they are grouped by level of priority. And it is defined to which stakeholder concrete requirement or group of requirements is assigned. But paper lack to proposition according to which criterion some requirement is corresponded to which priority level. And also there is no concrete recommendations how to distribute requirements between different types of stakeholders.

The work [Klimek, 2012] concerns gathering requirements and their formal verification using deductive approach. This approach is based on the semantic tableaux reasoning method and temporal logic.

Authors ground the necessity of developing of implementing formal methods and approaches for requirement analysis performing [Klimek, 2012].

Formal methods can constitute a foundation for providing natural and intuitive support for reasoning about system requirements and they guarantee a rigorous approach in software construction. The main motivation for this work is the lack of satisfactory and documented results of the practical application of deductive methods for the formal verification of requirement models [Klimek, 2012].

Temporal logic is a well established formalism for describing properties of reactive systems. It may facilitate both the system specifying process and the formal verification of non-functional requirements which are usually difficult to verify [Klimek, 2012].

The semantic tableaux method is quite intuitive and has some advantages over traditional deduction strategies. System requirements are gathered using some UML diagrams. Requirements engineering based on formal analysis and verification might play an essential role in producing the correct software since this approach increases reliability and trust in software. Deductive inference is always the most natural for human beings and is used intuitively in everyday life. A use case, its scenario and its activity diagram may be linked to each other during the process of gathering requirements. When activities and actions are identified in the use case scenario then their workflows are modeled using the activity diagram. The automation of this process is crucial and constitutes a challenge in the whole deductive approach [Klimek, 2012].

Diagram is decomposed on components. For every component it is proposed to use pattern allowing to describe in text operation, that match to it.

Also authors mentioned that temporal logic properties and formulas may be difficult to specify by inexperienced users and this fact can be a significant obstacle to the practical use of deduction-based verification tools.

Automatic transformation of workflow patterns to temporal logic formulas is proposed. These formulas constitute logical specifications of requirements models. The architecture of an automatic and deduction-based verification system is proposed. Authors expect that applying this concept results in the reduction of software development costs as some errors might be addressed in the software requirements phase and not in the implementation or testing phases. But analyzing complex software may cause too long formulas difficult for further processing [Klimek, 2012].

Paper [Teruel, 2011] proposes techniques for increasing of effectiveness communication processes between stakeholders. Authors represent three Goal-Oriented approaches, namely NFR framework, *i** and KAOS, are evaluated in order to determine which one is the most suitable to deal with this problem of requirements specification in collaborative systems. These Goal oriented approaches aimed to increase requirement modeling processes. *i** framework is complicated with several relations that are not presented in classical UML diagrams. KAOS goal model lets to define dependencies between goals. This classification was elucidated by *i** framework. Authors represented comparative analysis of considered frameworks and define their peculiarities.

*i** only provides a partial support for quantifying the relations among requirements when using contribution links. The *i** approach also fails in representing the requirements importance, giving no support to determine which requirements are more important than others [Teruel, 2011].

Nevertheless, the other two GO approaches also share this lack of representation of the importance of each requirement. KAOS also fails in the same features than *i** but, unlike this approach, KAOS obtains a lower (or the same) score in almost all features except for the Hierarchical Representation feature, thanks to its tree-based representation. Finally, the NFR framework is the less suitable approach, obtaining a very low score, because of both the lack of expressiveness to specify FRs and its lack of adaptability to represent Collaborative Systems Characteristics. Also approach how to analyze requirement specification to match requirements to proposed classification of goals and sub-goals do not identified [Teruel, 2011].

ANALYSIS OF EXISTING APPLICATION LIFE CYCLE PRODUCTS ON THE MARKET

There are many ALM tools available for tracking application changes. These range from dedicated ALM products that monitor an application from inception to completion, automatically sorting files into logical buckets as changes are noted, to simple wikis that require team members to record changes manually,

this section reviewed the most popular among them, namely: Visual Studio, IBM Rational Team Concert, FusionForge and Team Forge

1.1 Microsoft Visual Studio with Team Foundation Server

Microsoft Visual Studio (VS) is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web applications and web services.

Microsoft Visual Studio has the next features:

- allow to write and debugg code;
- has a forms and data Designer;
- allow calculate code metrics;
- has plugins support;
- support source control via Team Foundation Server or Git.

Microsoft Visual Studio is Integrated with Team Foundation Server (TFS) proposing common Microsoft product that provides the next functions:

- source code management (either with Team Foundation Version Control or Git);
- reporting;
- requirements management;
- project management (for both agile software development and waterfall teams);
- automated builds;
- lab management;
- testing;
- release management capabilities.

TFS is often used on large enterprises. Free version of the product is an IDE (has limitations in code editor, etc.) with minimal possibilities (version control system, class diagram, metrics calculation, etc.) for life cycle support. In turn, the paid version allows you to not only make full use of the code editor, but also almost fully support the life cycle of the developed software [Microsoft, 2015].

1.2 IBM Rational Team Concert

Rational Team Concert (RTC) is a software development team collaboration tool developed by the Rational Software brand of IBM, who first released it in 2008. The software is available in both client

versions and a Web version. It provides a collaborative environment that software development teams use to manage all aspects of their work. It has the next features:

- supports all main paradigms of a software development;
- allow to create local source control;
- defect tracking;
- build Management;
- has a customized dashboard;
- allow to tracking changes in items;
- has a report system for fast-tracking of detected defects.

As well as Microsoft Visual Studio, IBM Rational Team Concert is focused on large enterprises and the base package only allows you to plan, schedule and monitor progress of work, version control and bug/feature tracking. In order to design, system requirements analysis, etc. you need not buy separate IBM Rational products [IBM b), 2015].

1.3 FusionForge

FusionForge (FF) is a free software application descendant of the forge (web-based project-management and collaboration software) originally created for running the SourceForge.net platform. FusionForge is licensed under the GNU General Public License, and is a fork/renaming of the code which was previously named GForge.It has the next features:

- provides version control by using GNU arch, Bazaar, CVS, Darcs, Git or Subversion)
- allow to bug/feature-tracking;
- has own messaging feature, that can be deployed to run a self-hosted forge;
- allow to create surveys for users and admins;
- plugins support;
- allow to create wiki for systems;
- has a task management system.

FusionForge well suited for medium and small teams. Environment allows automating some of the life cycle processes, flexibly configuring the environment for development team style (e.g. setting of fields to keep track of bugs). Of the downsides can highlight the need for a local server to work with the environment (in case of problems with the server team will not not have an access to the tasks and

repositories, due to lack of a desktop client), also environment does not provides tools for software design (e.g. UML support) [Fushionforge, 2016].

1.4 TeamForge

TeamForge (TF) is a collaborative revision control and software development management system. It provides a front-end to a range of software development lifecycle services and integrates with a number of free software / open source software applications (such as PostgreSQL and Subversion).It has the next features:

- has a revision control;
- software development management system;
- bug tracking system;
- allows you to track the progress of performed work;
- allows you to keep track of tasks after release (bug/feature completion tracking);
- allows to create any discussions for platform users;
- can be integrated in other software (Visual Studio, Microsoft Office, Eclipse and so on).

This environment is mainly focused on large companies that use additional software in the development of programs (for example Microsoft Office, Visual Studio, Outlook etc.) and its own servers (for deploying lifecycle environment). But, this environment, does not allow to build UML diagrams (does not support design process).

Table 1 represents results of comparison analysis of ALM environments [Teamforge, 2016].

As you can see from the Table 1 listed lifecycle management environments has both advantages and disadvantages. Disadvantages of considered ALMs are the next:

- need to be purchased (Visual Studio, IBM Rational Concert, TeamForge);
- not available on other operating systems (Visual Studio);
- lack of support of the design process (FusionForge, TeamForge, IBM Rational Concert (only in additional packages);
- lack of software testing tools (FusionForge, TeamForge, IBM Rational Concert (only in additional packages);
- lack of code editor (FusionForge, TeamForge, IBM Rational Concert (only in additional package));
- no desktop client (FusionForge, TeamForge);
- attachment only to the local server (FusionForge, TeamForge).

Table 1. Comparison of ALM environments

	VS	FF	TF	RTC
Version control	+	+	+	+
Requirements management	Only in TFS	+	+	+
Bug/feature-tracking	Only in TFS	+	+	+
Plug-in support	+	+	-	Can only integrates in others IBM's products
Code editor	+	-	-	+
Code debugger	+	-	-	+
GUI designer	+	-	-	-
UML support	Partially	-	-	In Rational Software Architect
Code generation from models	+	-	-	+
Task management	+	+	+	+
Unit Testing	+	-	-	In Rational Functional Tester
Functional testing	-	-	-	In Rational Functional Tester
Generation of developer's documentation	-	-	-	-
Cross-platform	-	+	+	+
Free to Use	Only Community version (Limited version)	-	+	-

Task and challenges

Analyzing review of papers and tools formulate requirements for designed ALM environment formulate requirements to future system:

- support software developing at all stages of the life cycle;
- it must be cross-platform;
- can be deployed on local computer;
- contain modules supporting stakeholders collaboration.
- support following functions:
  - task managing
  - defining requirements;
  - designing new software by using a UML diagrams;
  - implementing a new software by using Eclipse IDE, UML to code convertor etc.;
  - testing developed software;
  - maintenance developed software (automatic creation of developers documentation, bug tracking list);
  - and monitor changes in the project with the help of git version control).

Proposed approach

Designing good application life cycle management environment should consider possibilities to change performing of some processes by means of defining proper configuration of plugins.

To simplify software development lifecycle management we use Eclipse platform that anbles to plug a variety of free plug-ins, among which there are plug-ins for software lifecycle management.

The Figure 1 containing a graphical representation of plug-ins that can be used on the phases of a typical software development life cycle (SDLC).

Represent description of chosen plug-ins: functionality:

- UML to Java Generator extends functionality of UML Designer – allows to generate java code from UML Class Diagrams;
- WindowBuilder extends functionality of Eclipse IDE – allows to build GUI;
- Eclipse Color Theme extends the capabilities of the Eclipse IDE code editor and Data Tools Platform sql editor;
- JUnit extends functionality of Eclipse IDE – allows to write Unit tests to Java classes;

- STAN extends functionality of Eclipse IDE – allows to analyze code that was written in Eclipse IDE;
- FindBugs extends functionality of Eclipse IDE – allows to find possible bottlenecks, errors and bugs in code;
- Jubula extends functionality of Eclipse IDE – allows to write and test program that was written in Eclipse IDE;
- JavaDoc extends functionality of Eclipse IDE – allows to write developer's documentation for the written code in Eclipse IDE.

Data flow between environment components and software development artifacts is represented on Figure 2.

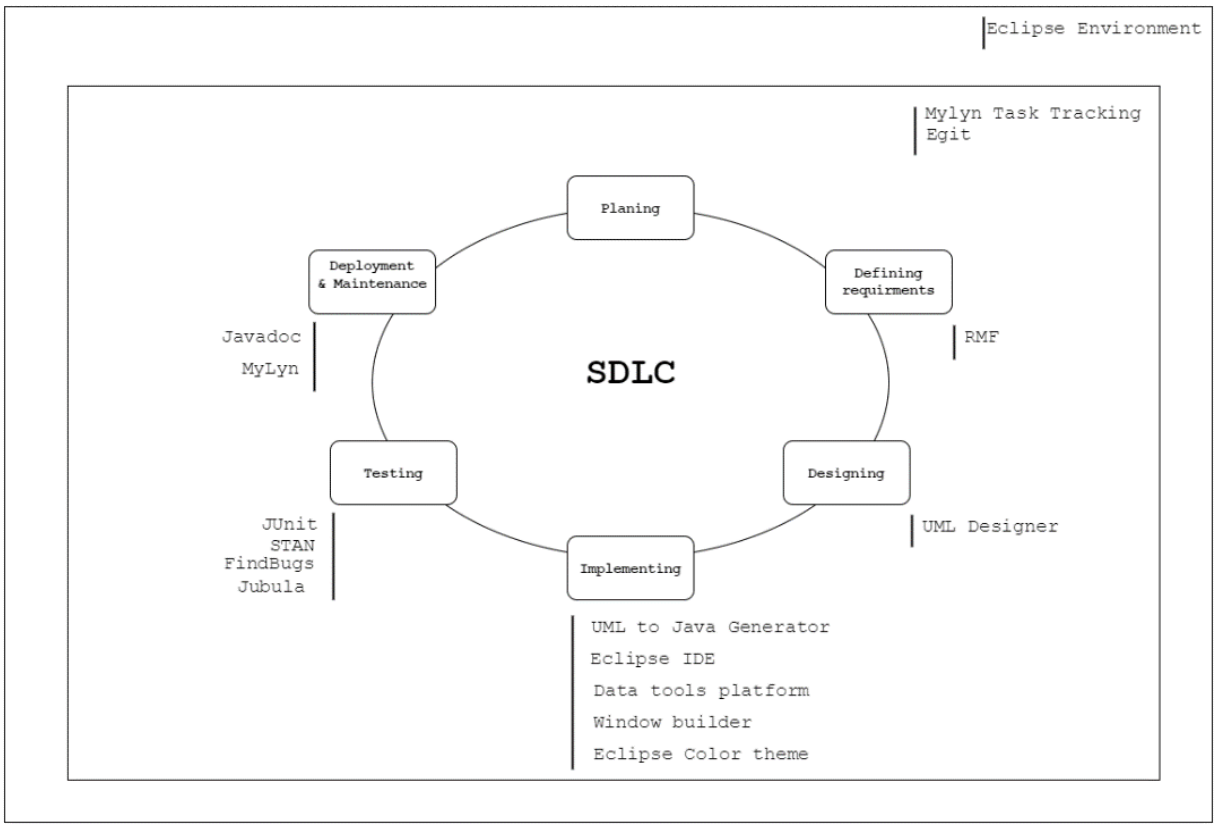


Figure 1. Plug-ins collaboration used to design a typical software development

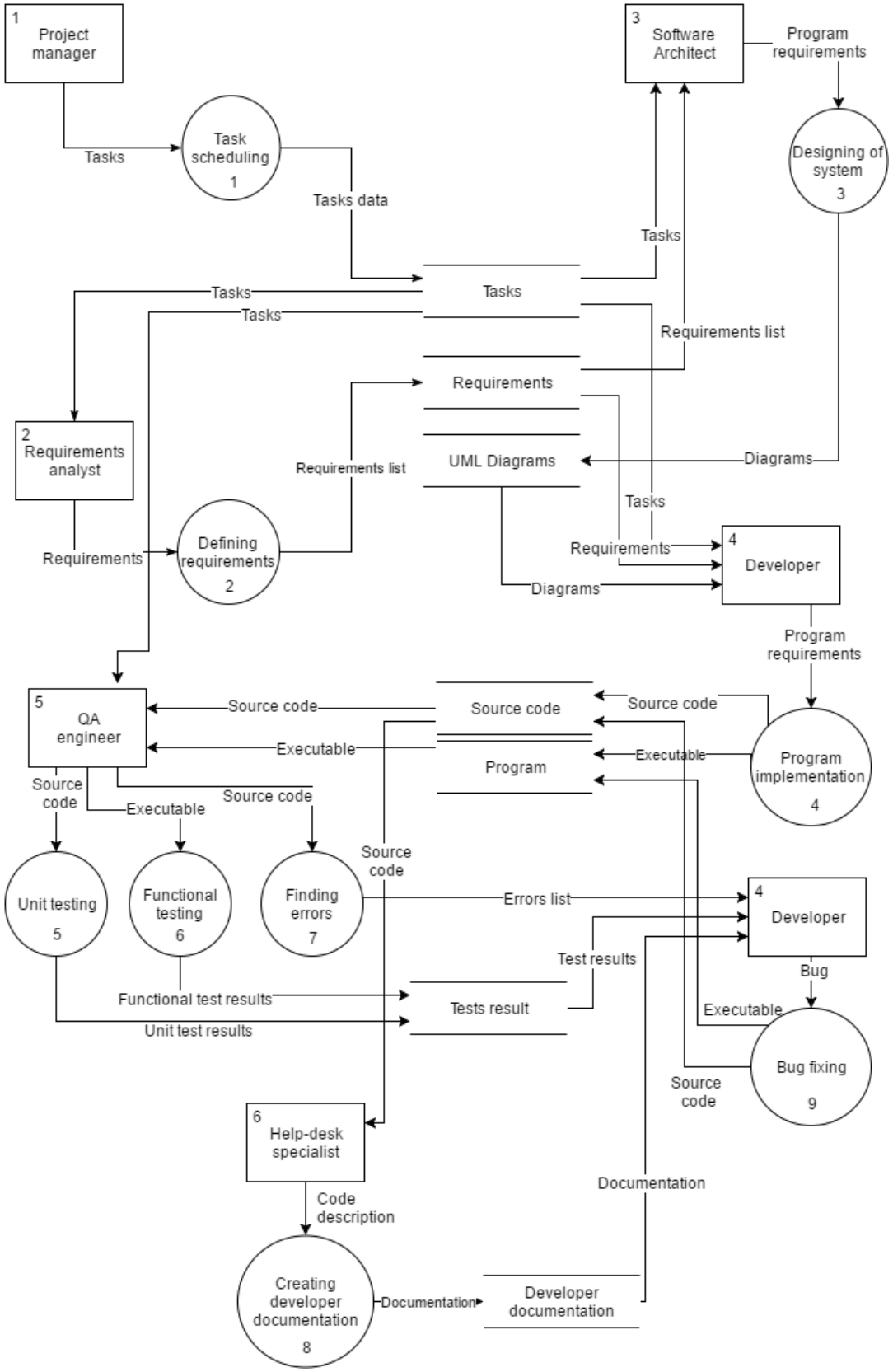


Figure 2. Data flow between ALM environment components

Represent Data flow between environment components

1. Project manager uses Mylyn for task scheduling;
2. From Mylyn Requirements analyst receive tasks for completion and by using RMF creates a requirements list;
3. Software architect receives tasks from Mylyn and Requirements list from EGit creates UML diagrams by using UML Designer;
4. Developer receives tasks from Mylyn, Requirements list and UML diagrams from EGit implements program in Eclipse IDE;
5. QA engineer receives tasks from Mylyn and source code from EGit creates Unit tests and perform Unit tests in JUnit;
6. QA engineer receives tasks from Mylyn and executable from EGit creates Functional in Jubula;
7. QA engineer receives tasks from Mylyn and source code from EGit perform Finding error process in STAN and FindBugs;
8. Help-desk specialist receives tasks from Mylyn and source code from EGit creates Developer documentation in JavaDoc
9. Developer receives error list, test results and developer documentation from EGit fixes bug in Eclipse IDE;
10. Tasks, Requirements, UML Diagrams, Source code, Program, Tests result and Developer documentation are stores in EGit.

Schema, representing matching with plug-ins and stakeholders' activities is represented in figure 3.

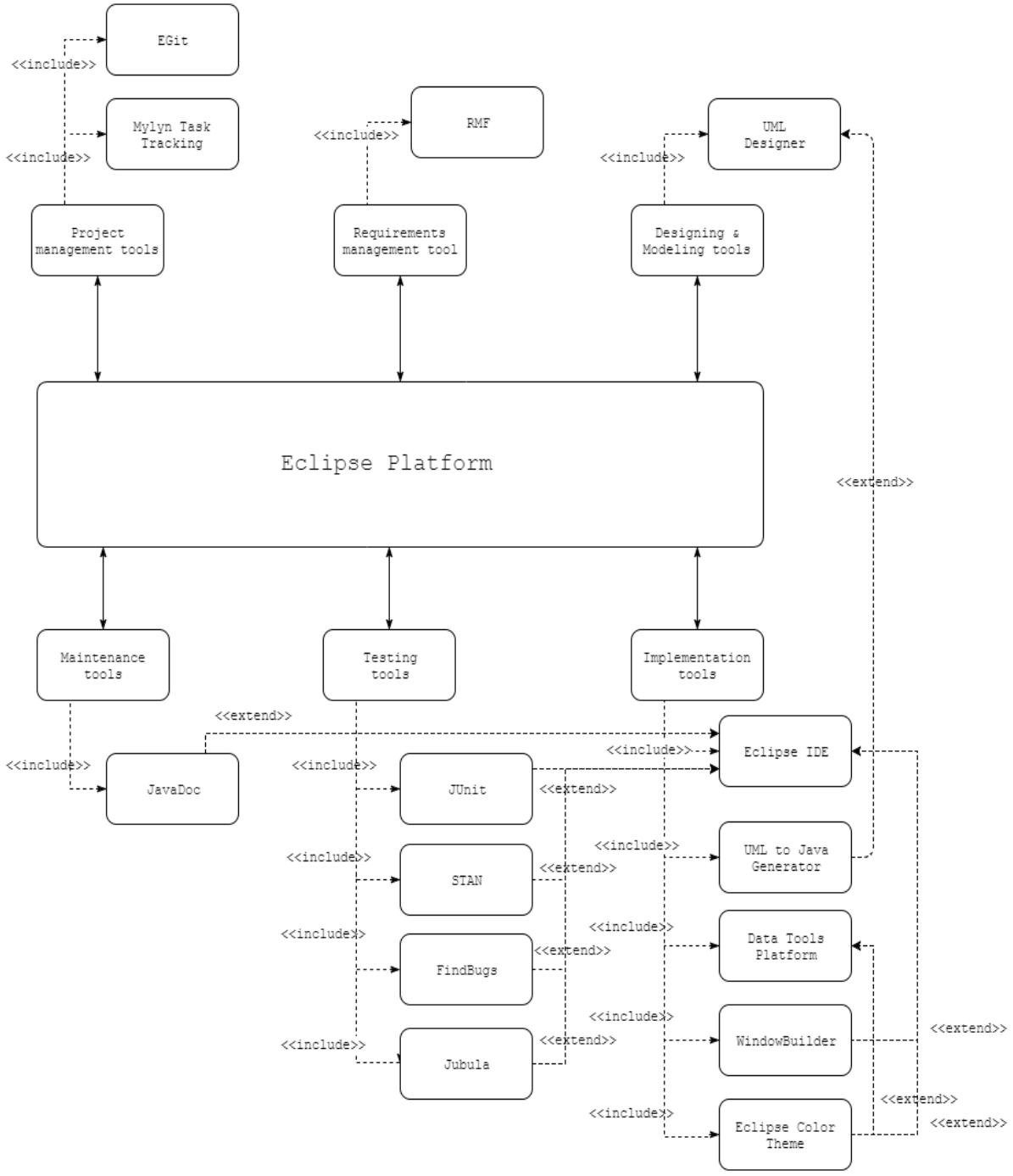


Figure 3. ALM environment and plug-ins interaction scheme

Work processes for each stakeholder role

Analyzing schemas on Figures 1-3 analyze work processes for each stakeholder role.

Project manager activities

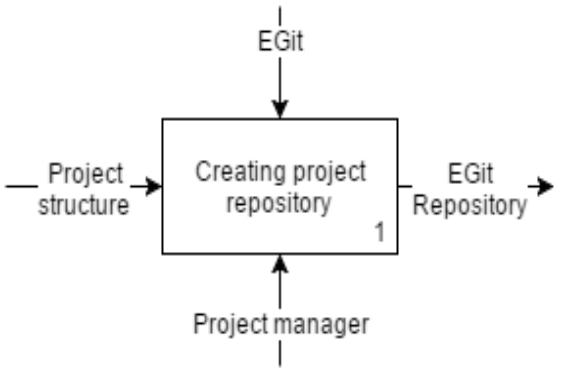


Figure 4. Creating project repository

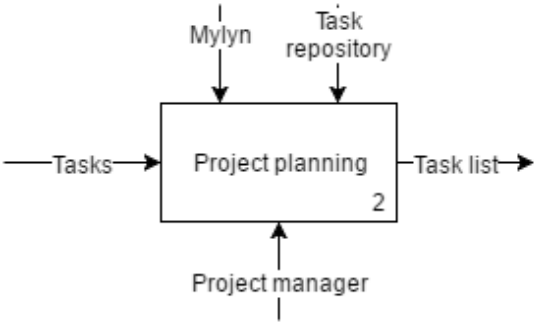


Figure 5. Project Planning

Requirement analyst activities

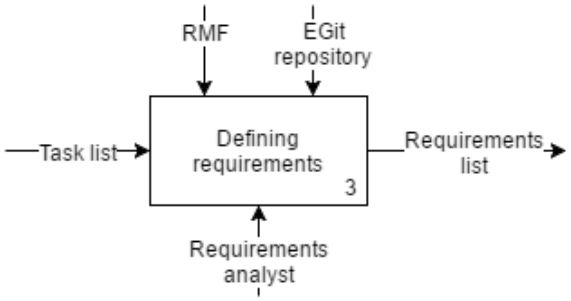


Figure 6. Defining requirements

Software architector activities

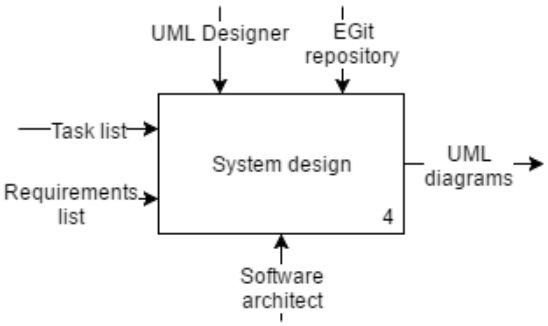


Figure 7. System designing

Software developer activities

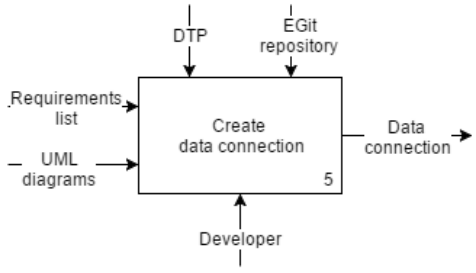


Figure 8. Create data connection

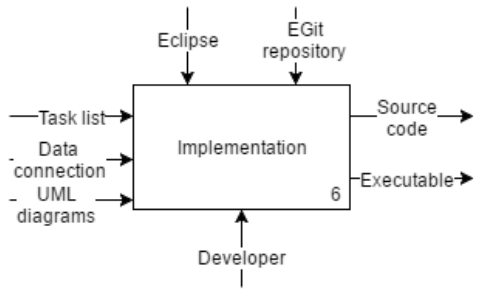


Figure 9. Upload(download) Implementation

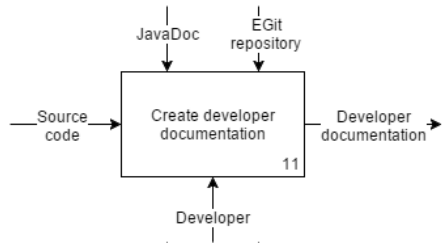


Figure 10. Create developer documentation

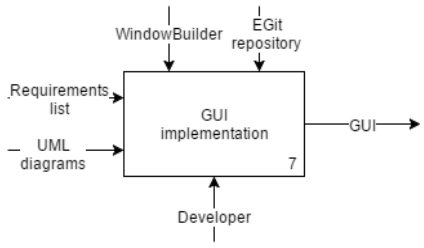


Figure 11. Upload(download) GUI Implementation

Activities of QA Engineer

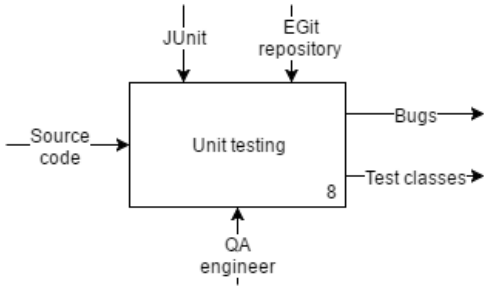


Figure 12. Unit testing

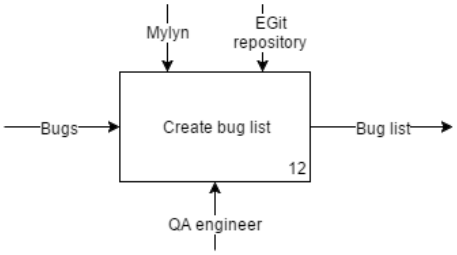


Figure 13. Create bug-list

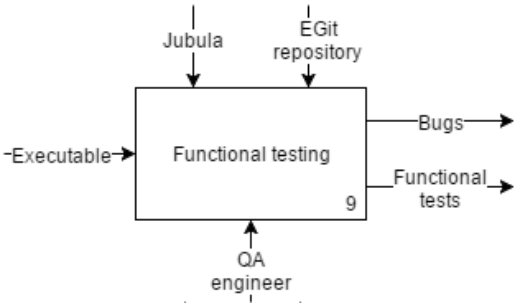


Figure 14. Functional testing

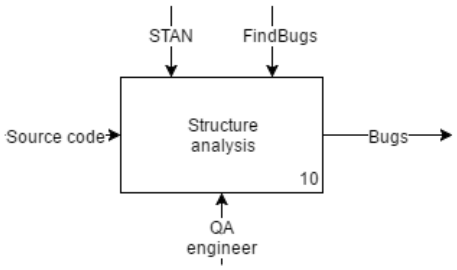


Figure 15. Structure analysis

Activities of Help-desk specialist

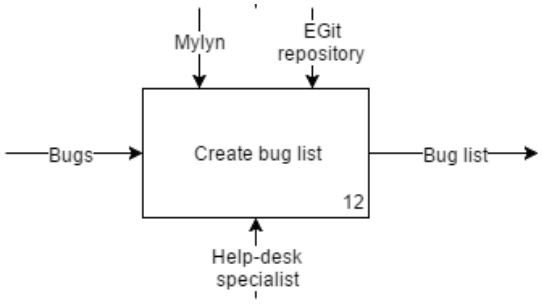


Figure 16. Create bug list

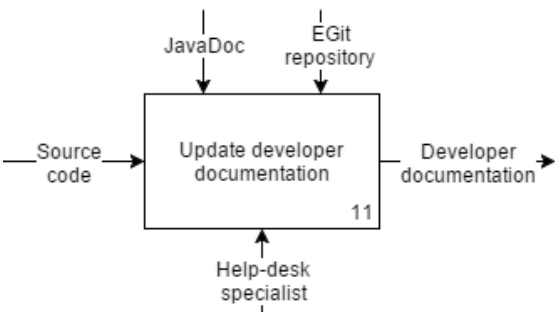


Figure 17. Update developer documentation

Conclusions

Cross-platform environment for supporting ALM that is based on Eclipse is proposed in this paper. It satisfies to all challenges, formulated above, namely cross-platform support; free; extensible by means of setting flexible plug-ins configuration; supporting extensible stakeholders activities and organizing different data flows between modules by means of adding new plugins; can be configured on local computer; also can be deployed on server for stakeholders interconnection. Represented activities according stakeholders roles are focused on actions, performed by stakeholders. In order to adopt software development life cycle processes to needs of specific enterprise sequence of actions, and consequently configuration of plugins can change. As eclipse provides supporting desktop, cloud and platforms IDEs, proposed ALM can be integrated with various programming languages, for example by means of creating developing perspectives [Eclipce, 2015].

Implementing of designed ALM will allow reducing the time required to develop new software, will increase the involvement of the team in the development process. In this turn, project managers will be able to track the progress of the project and identify the risks of disrupting the work schedule.

With the advent of such software environments in various industries, software developers and other stakeholders will spend less time on development, which in turn will reduce the cost of development.

Further research

To propose an ALM approach, facilitating requirement analysis and software designing; focusing on Model-Driven techniques, making accent on code generation techniques.

Bibliography

[Chebanyuk and Markov, 2016] Chebanyuk E. and Markov K. (2016). An Approach to Class Diagrams Verification According to SOLID Design Principles.In Proceedings of the 4th International Conference on Model-Driven Engineering and Software Development - Volume 1: MODELSWARD, ISBN 978-989-758-168-7, pages 435-441. DOI: 10.5220/0005830104350441 http://www.scitepress.org/DigitalLibrary/PublicationsDetail.aspx?ID=HASwCJGMcXc=&t=1

[Chebanyuk, 2014] Chebanyuk, Elena.2014. Method of behavioural software models synchronization. International journal Informational models and analysis. – 2014, №2 P 147-163 http://www.foibg.com/ijima/vol03/ijima03-02-p05.pdf

[Eclipce, 2016] https://eclipse.org/ide/

[Escande et al., 2013] Loup Escande E. and Christmann O. (2013). Requirements Prioritization by End-users and Consequences on Design of a Virtual Reality Software - An Exploratory Study.In Proceedings of the 8th International Conference on Evaluation of Novel Approaches to Software Engineering - Volume 1: ENASE, ISBN 978-989-8565-62-4, pages 5-14. DOI: 10.5220/0004397900050014

[Filho et al., 2016] Filho A., do Prado H. and Ferneda E. (2016). A Metadata-based Architecture for Identification and Discovery of Services in SOA..In Proceedings of the 18th International Conference on Enterprise Information Systems - Volume 2: ICEIS, ISBN 978-989-758-187-8, pages 298-305. DOI: 10.5220/0005867702980305

[FusionForge, 2016] https://fusionforge.org/

[Grichi et al., 2015] Grichi H., Mosbahi O. and Khalgui M.. ROCL: New Extensions to OCL for Useful Verification of Flexible Software Systems. DOI: 10.5220/0005522700450052 In Proceedings of the 10th International Conference on Software Engineering and Applications (ICSOFT-EA-2015), pages 45-52 ISBN: 978-989-758-114-4

[IBM b), 2015] https://jazz.net/library/article/632

[IBM, 2015] http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=APW12347USEN

[Inoue et al., 2015] Inoue W., Hayashi S., Kaiya H. and Saeki M.. Multi-dimensional Goal Refinement in Goal-Oriented Requirements Engineering.DOI: 10.5220/0005499301850195 In Proceedings of the 10th International Conference on Software Engineering and Applications (ICSOFT-EA-2015), pages 185-195 ISBN: 978-989-758-114-4

[Klimek, 2012] Klimek R. (2012). Proposal to Improve the Requirements Process through Formal Verification using Deductive Approach.In Proceedings of the 7th International Conference on Evaluation of Novel Approaches to Software Engineering - Volume 1: ENASE, ISBN 978-989-8565-13-6, pages 105-114. DOI: 10.5220/0004001901050114

[Microsoft, 2015] https://www.visualstudio.com/tfs/

[Misra, 2017] Harekrishna Misra Managing User Capabilities in Information Systems Life Cycle: Conceptual Modeling. International Journal of Information Science and Management Vol. 15, No. 1, 2017, 39-58 http://ijism.ricest.ac.ir/index.php/ijism/article/view/936

[OMG, 2006] ftp://ftp.omg.org/pub/presentations/ajw_alm/ALM.pdf

[Sharma et al., 2014] Sharma R., Gulia S. and Biswas K. (2014). Automated Generation of Activity and Sequence Diagrams from Natural Language Requirements.In Proceedings of the 9th International

Conference on Evaluation of Novel Approaches to Software Engineering - Volume 1: ENASE, ISBN 978-989-758-030-7, pages 69-77. DOI: 10.5220/0004893600690077

[Stanford, 2016] http://nlp.stanford.edu/software/stanford-dependencies.shtml

[Teamforge, 2016] https://www.collab.net/products/teamforge-alm

[Teruel et al., 2011] Teruel M., Navarro E., López-Jaquero V., Montero F. and González P. (2011). A COMPARATIVE OF GOAL-ORIENTED APPROACHES TO MODELLING REQUIREMENTS FOR COLLABORATIVE SYSTEMS.In Proceedings of the 6th International Conference on Evaluation of Novel Approaches to Software Engineering - Volume 1: ENASE, ISBN 978-989-8425-57-7, pages 131-142. DOI: 10.5220/0003466301310142

Authors' Information

Elena Chebanyuk *– Assoc. Prof. of Software Engineering Department, National Aviation University, Kyiv, Ukraine,*
Major Fields of Scientific Research: Model-Driven Architecture, Model-Driven Development, Software architecture, Mobile development, Software development,
e-mail: chebanyuk.elena@ithea.org

Oleksii Hlukhov *– student of Software Engineering Department, National Aviation University, Kyiv, Ukraine.*

IN MEMORIAM:

Prof. Volodimir Stepanovich Donchenko

(1947-2017)

Mathematician (Kiev University, 1970), Dr. Sci.(2007), Professor (1992)

He had worked in the Faculty of Cybernetics of Kiev National University, Ukraine, from 1973:

- Head. of Dep. of Mathematics in Military Air Defense Academy of Ukraine (1987-93)
- Head. Dep. of Applied Statistics (1998-2002);
- Professor of Dep. of System's analysis and decision making theory (from 2002);
- Deputy Dean of Scientific work (1995-2001) .

His main scientific achievements are in the Probability theory, Mathematical modeling, Algebraic problems of pattern recognition.

He had developed the Mathematical Hough Transform Theory (see: Donchenko V.S. The Hough Transform and uncertainty // Intern. J. on Information Theories and Applications.–2003.–10, N 4. – pp. 376–379. http://www.foibg.com/ijita/vol10/ijita10-4-p03.pdf).

In 2007, Prof. Donchenko had been awarded by the International Prize "ITHEA" for his great achievements in the field of information theories and applications.

Prof. Donchenko will be deeply missed!

ITHEA ISS

TABLE OF CONTENTS

Comparison Software Systems Based on Information Quality Measuring
Krassimir Markov, Krassimira Ivanova, Stefan Karastanev 103
Application of Biosensors for Plants Monitoring
Oleksandr Palagin, Volodymyr Grusha, Hanna Antonova, Oleksandra Kovyrova, Vasyl Lavrentjev 115
Toward Measuring Linguistic Complexity: Grammatical Homonymy in The Russian Language
Olga Nevzorova, Alfiya Galieva, Vladimir Nevzorov 127
Use of At-Technology Workbench for Construction of Tutoring Integrated Expert Systems
Galina V. Rybina, Victor M. Rybin, Yuri M. Blohin, Elena S. Sergienko 141
Research on the Property "Avalanche Effect" in IDA Cryptographic Algorithm
Ivan Ivanov, Stella Vetova, Krassimira Ivanova, Neli Maneva 150
Inductive Modeling Method GMDH in the Problems of Data Mining
Yuriy Zaychenko, Galib Hamidov 156
Cross-Platform Environment for Application Life Cycle Management
Elena Chebanyuk, Oleksii Hlukhov 177
In memoriam: Prof. Volodimir Stepanovich Donchenko (1947-2017) 199
Table of contents 200